Chapter 4.
Hybrid tree and local search

Search strategies for visiting nodes

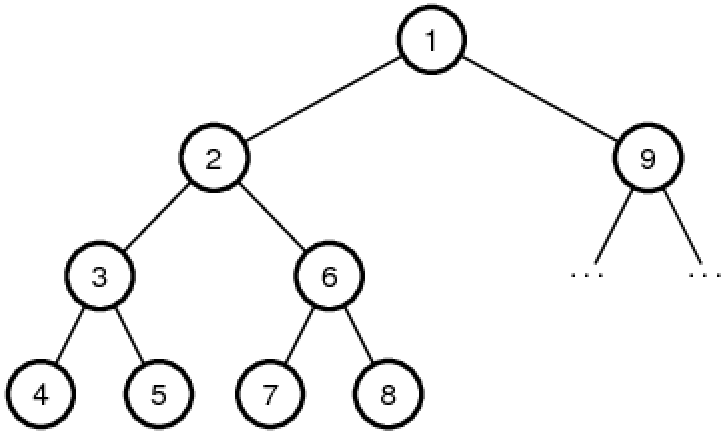
Variable ordering exploiting the structure

Value ordering used in partial search strategy

Variable neighborhood search

DFS

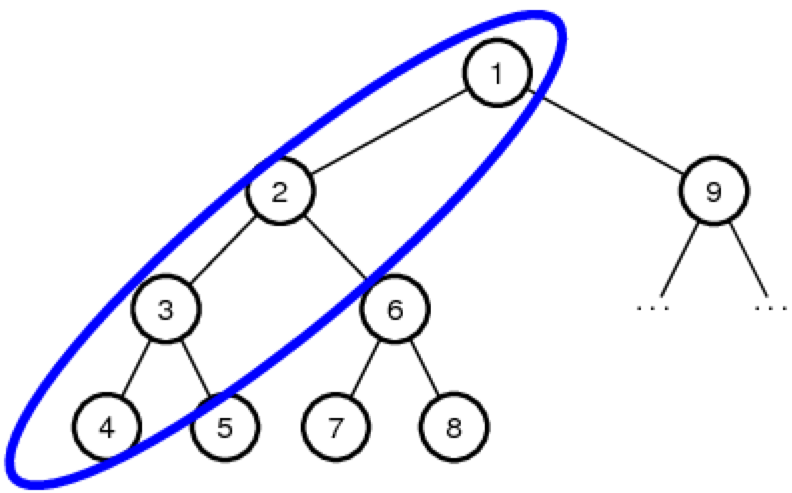
Depth First



DFS

Depth First
Advantages

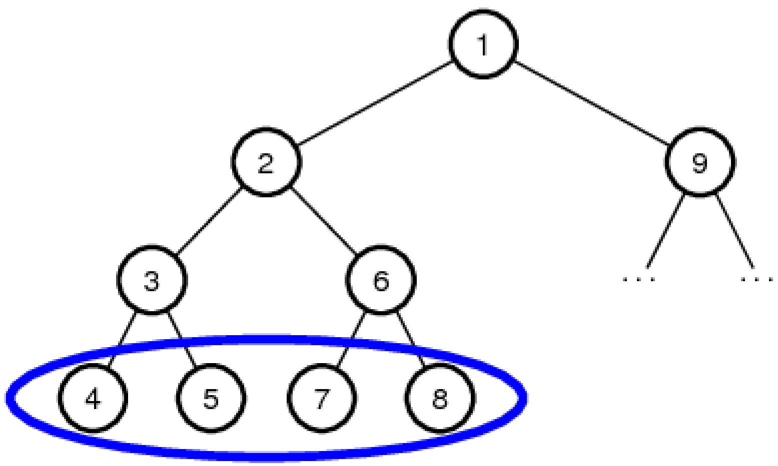
- Incrementality



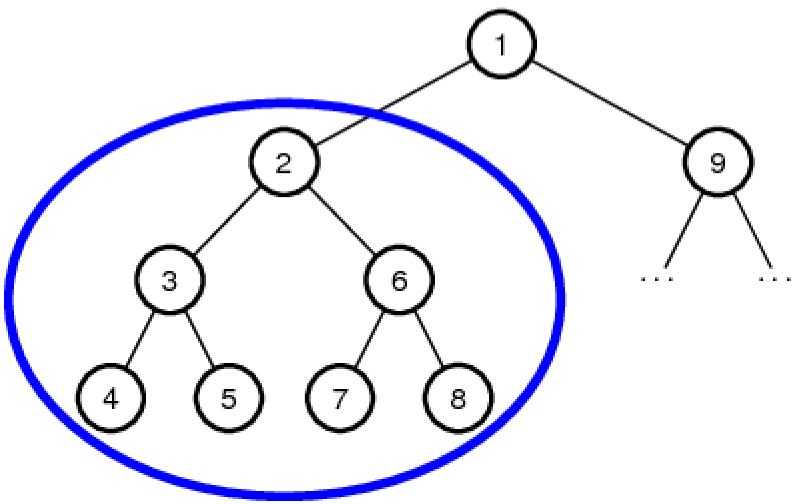
DFS

Depth First
Advantages

- Incrementality
- Anytime (sort of)



DFS



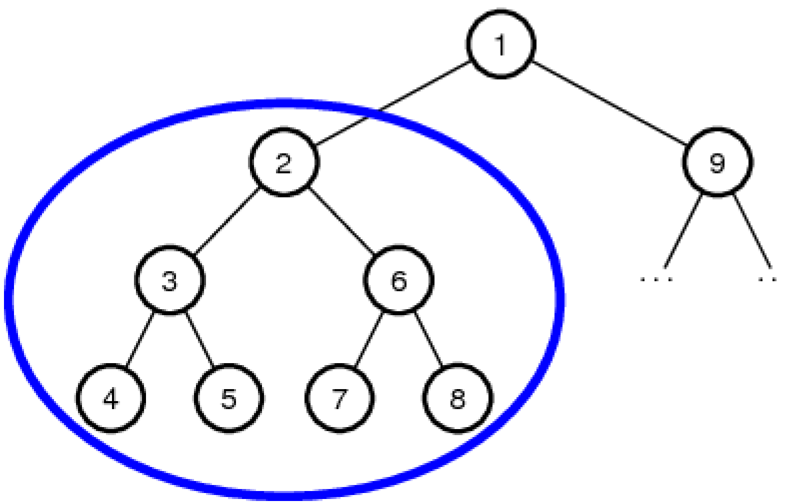
Depth First
Advantages

- Incrementality
- Anytime (sort of)

But

- Thrashing

DFS



Depth First
Advantages

- Incrementality
- Anytime (sort of)

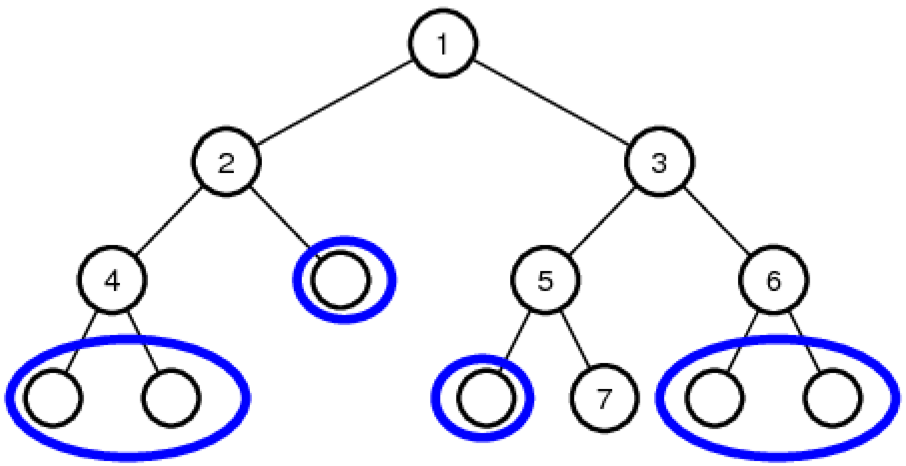
But

- Thrashing
- No global lower bounds

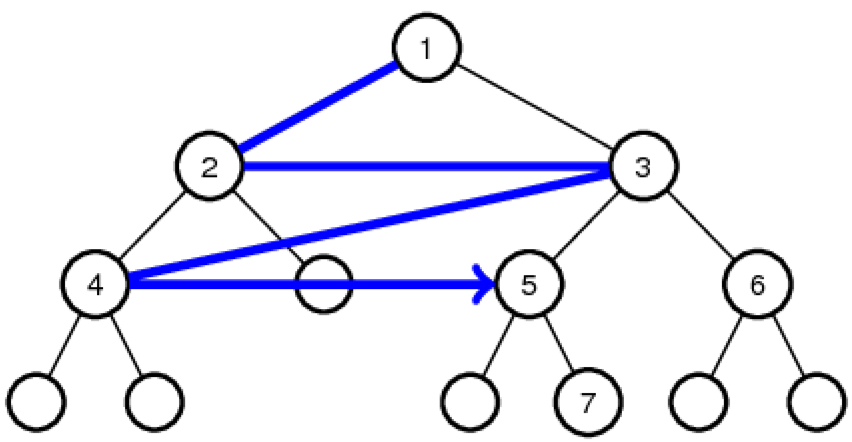
BFS

Best first

- Memory requirements



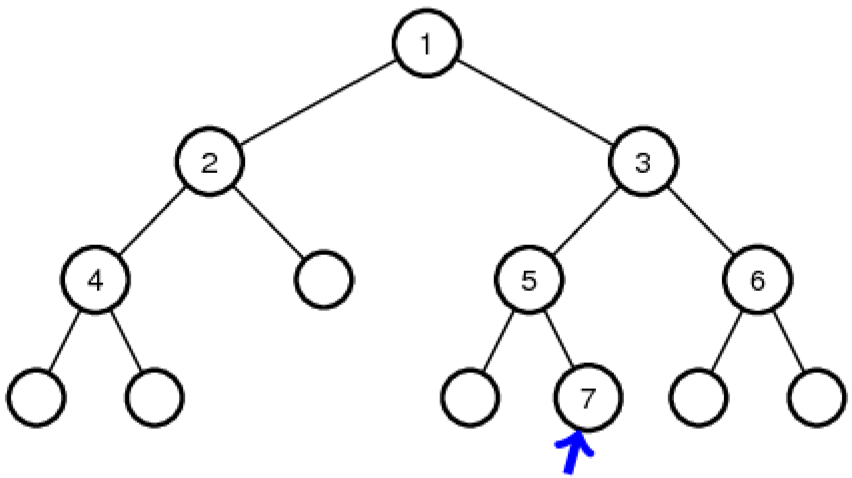
BFS



Best first

- Memory requirements
- No incrementality or even greater memory cost

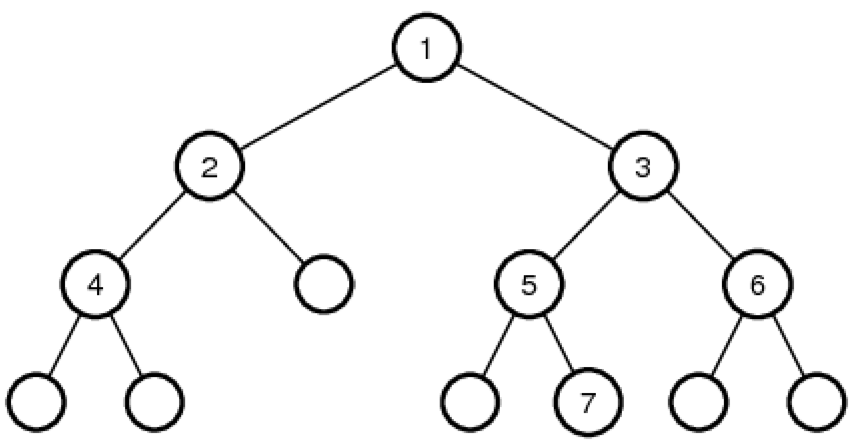
BFS



Best first

- Memory requirements
- No incrementality or
  even greater memory cost
- Not anytime

BFS



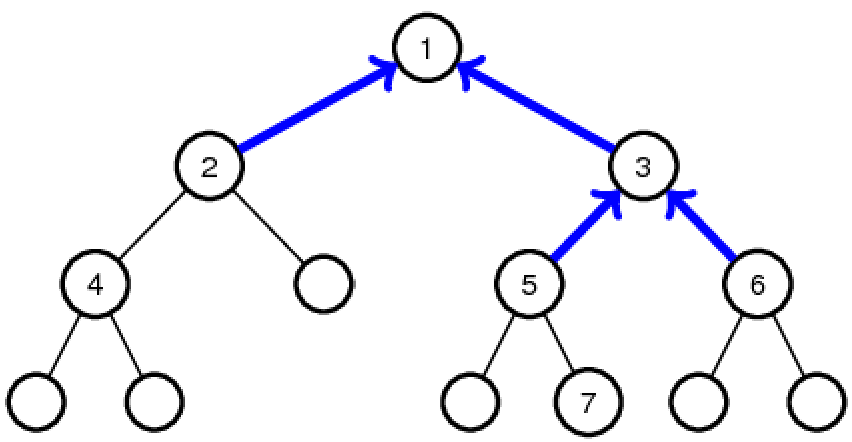
Best first

- Memory requirements
- No incrementality or even greater memory cost
- Not anytime

but

- Theoretical guarantees

BFS



Best first

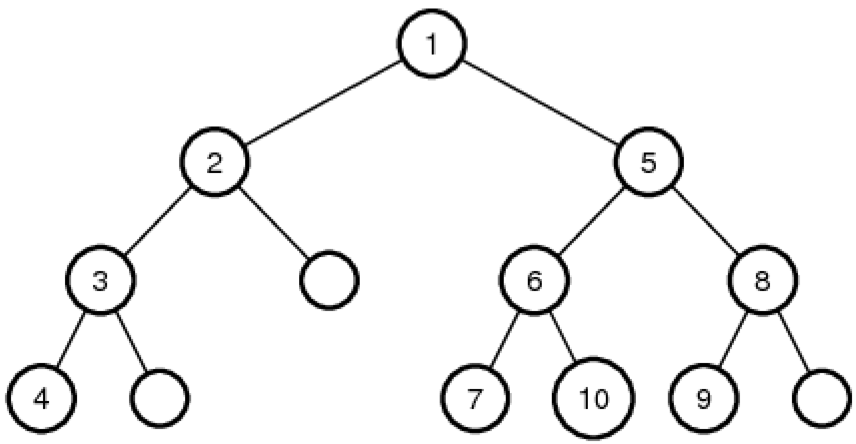
- Memory requirements
- No incrementality or even greater memory cost
- Not anytime

but

- Theoretical guarantees
- Global lower bounds

HBFS

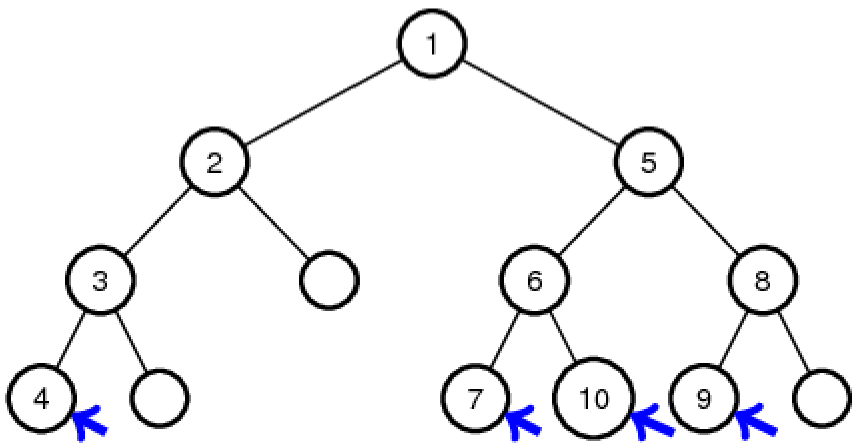
BFS with DFS probes*



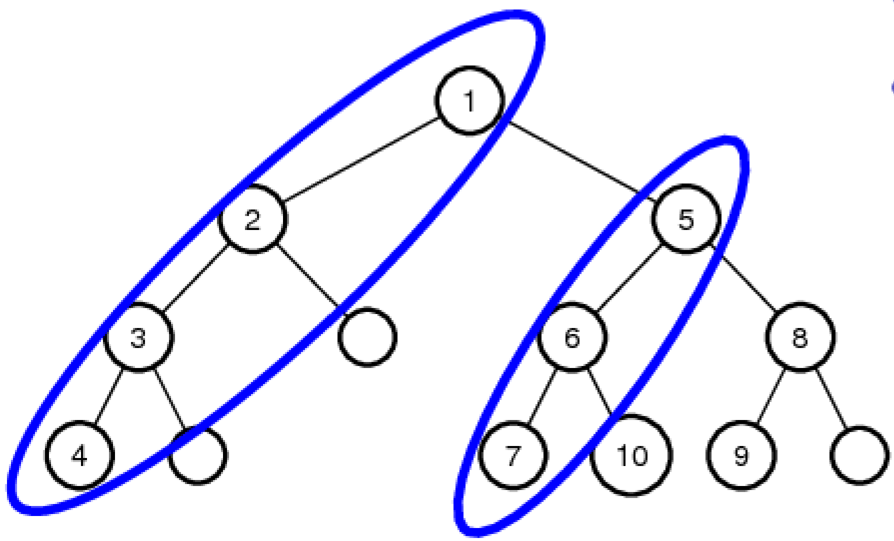
HBFS

BFS with DFS probes*

- Improved anytime behavior



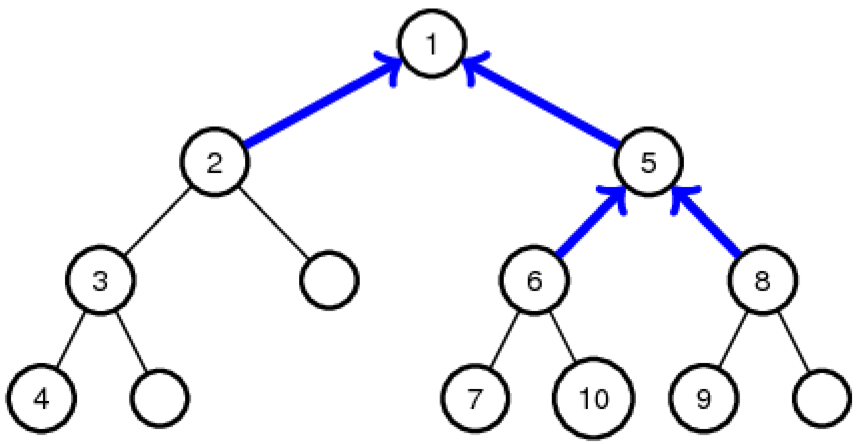
HBFS



BFS with DFS probes*

- Improved anytime behavior
- Incrementality
  without memory overhead

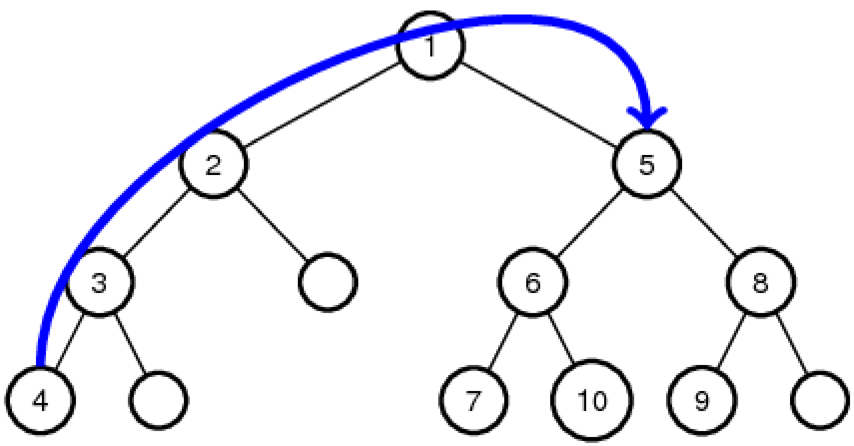
HBFS



BFS with DFS probes*

- Improved anytime behavior
- Incrementality
  without memory overhead
- Lower bounds

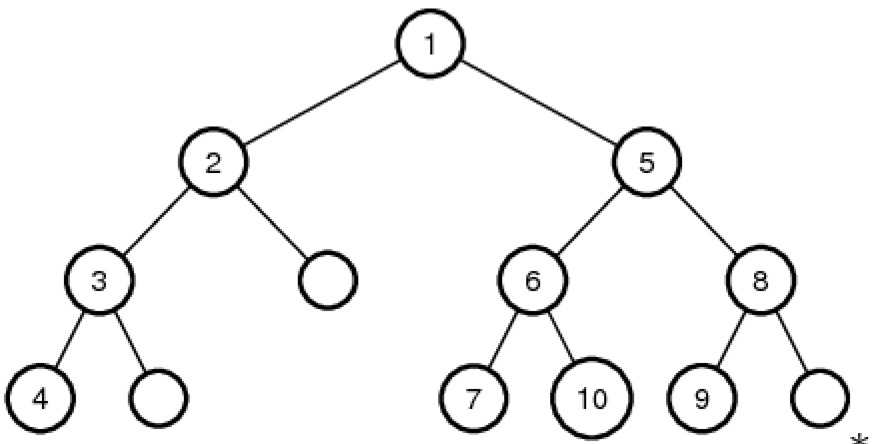
HBFS



BFS with DFS probes*

- Improved anytime behavior
- Incrementality
  without memory overhead
- Lower bounds
- Some of the
  advantages of restarting

HBFS



BFS with DFS probes*

- Improved anytime behavior
- Incrementality
  without memory overhead
- Lower bounds
- Some of the
  advantages of restarting

* With
adaptive heuristic for probe size

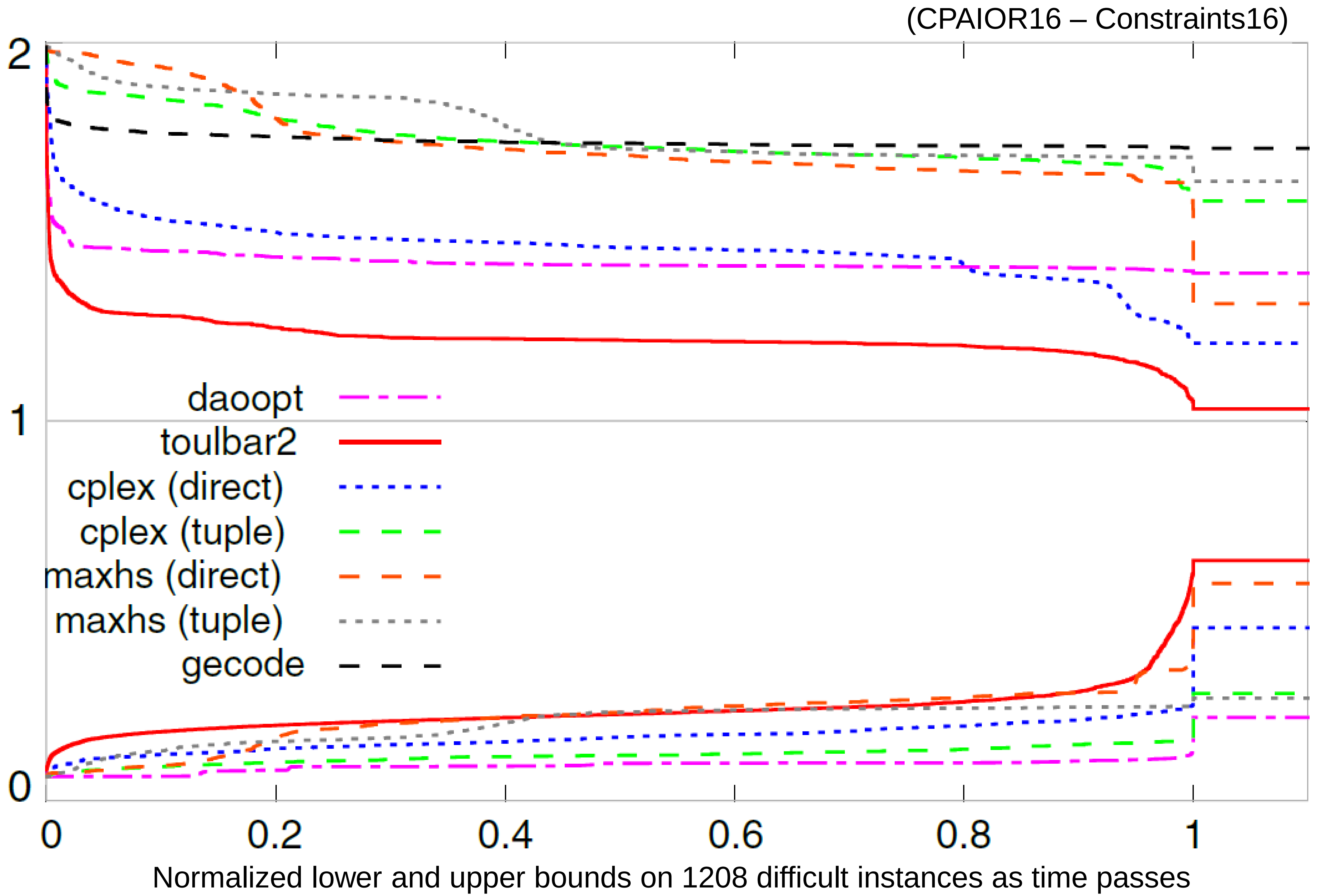
Benchmark

- MRF: Probabilistic Inference Challenge 2011 (uai format)
- CVPR: Computer Vision and Pattern Recognition OpenGM2 (uai)
- CFN: MaxCSP 2008 Competition and CFLib (wcsp format)
- WPMS: Weighted Partial MaxSAT Evaluation 2013 (wcnf format)
- CP: MiniZinc Challenge 2012 & 2013 (minizinc format)

Number of instances and their total compressed (gzipped) size:

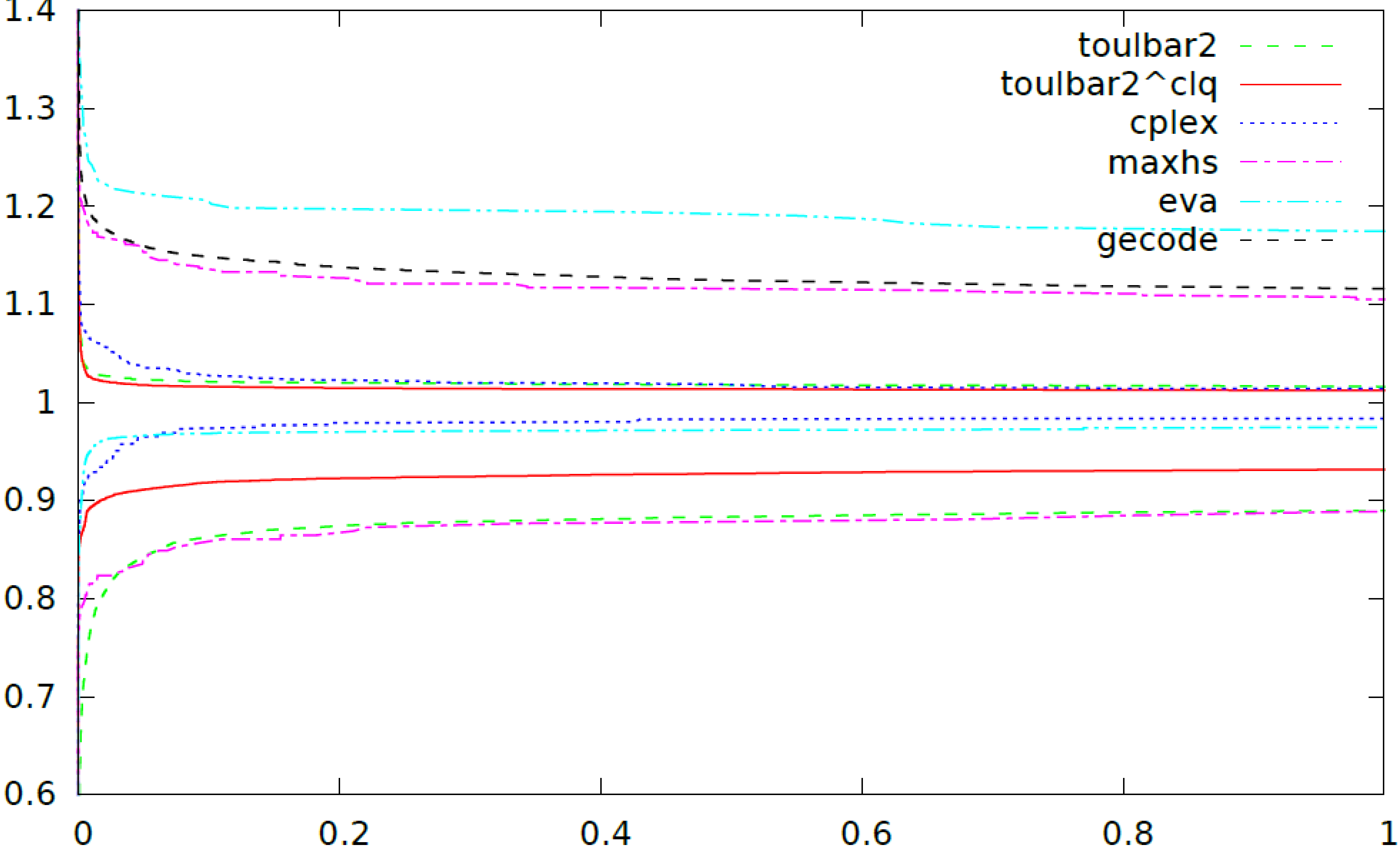
Benchmark	Nb.	UAI	WCSP	LP(direct)	LP(tuple)	WCNF(direct)	WCNF(tuple)	MINIZINC
MRF	319	187MB	475MB	2.4G	2.0GB	518MB	2.9GB	473MB
CVPR	1461	430MB	557MB	9.8GB	11GB	3.0GB	15GB	N/A
CFN	281	43MB	122MB	300MB	3.5GB	389MB	5.7GB	69MB
MaxCSP	503	13MB	24MB	311MB	660MB	73MB	999MB	29MB
WPMS	427	N/A	387MB	433MB	N/A	717MB	N/A	631MB
CP	35	7.5MB	597MB	499MB	1.2GB	378MB	1.9GB	21KB
Total	**3026**	0.68G	2.2G	14G	18G	5G	27G	1.2G

http://genoweb.toulouse.inra.fr/~degivry/evalgm



(CPAIOR16 – Constraints16)
2
1
0
daoopt
toulbar2
cplex (direct)
cplex (tuple)
maxhs (direct)
maxhs (tuple)
gecode
0
0.2
0.4
0.6
0.8
1
Normalized lower and upper bounds on 1208 difficult instances as time passes

Results exploiting cliques (CP17)



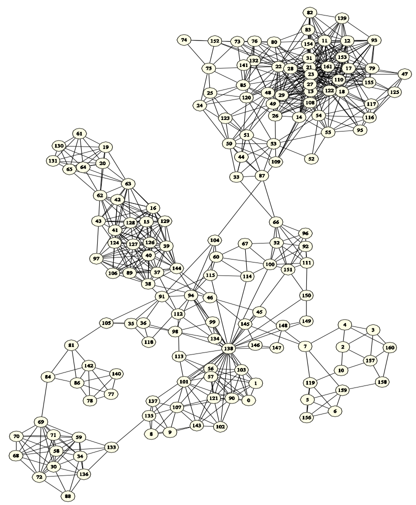
Normalized lower and upper bounds on 252 instances as time passes

Bibliography

- For benchmarking and solvers comparisons, see *Multi-Language Evaluation of Exact Solvers in Graphical Model Discrete Optimization,* Hurley et al., *Constraints 2016*.
- For hybrid search, see *Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP,* Katsirelos et al., *CP2015*.

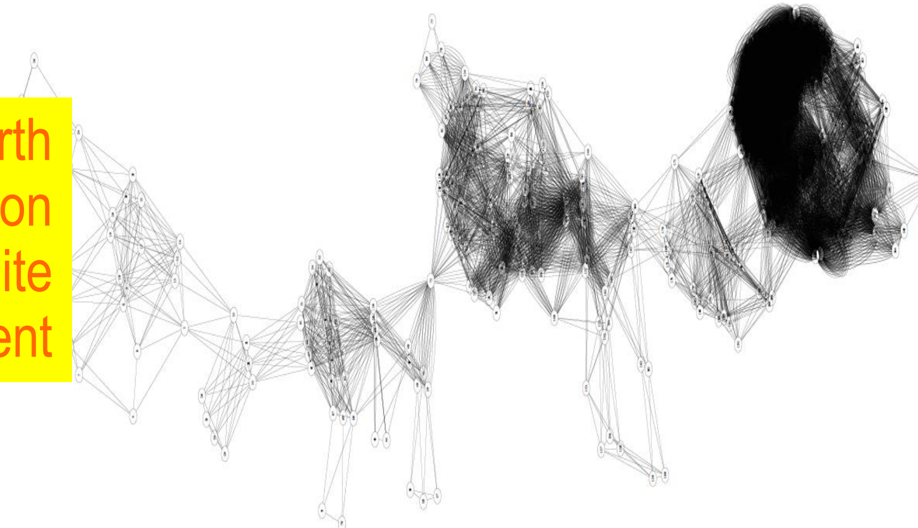
Many real applications have a structured network

Radio
Link
Frequency
Assignment



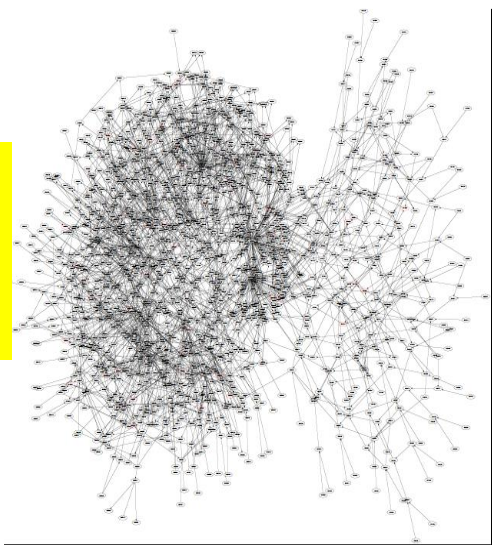
CELAR SCEN-07r
(*Constraints* 4(1), 1999)

Earth
Observation
Satellite
Management



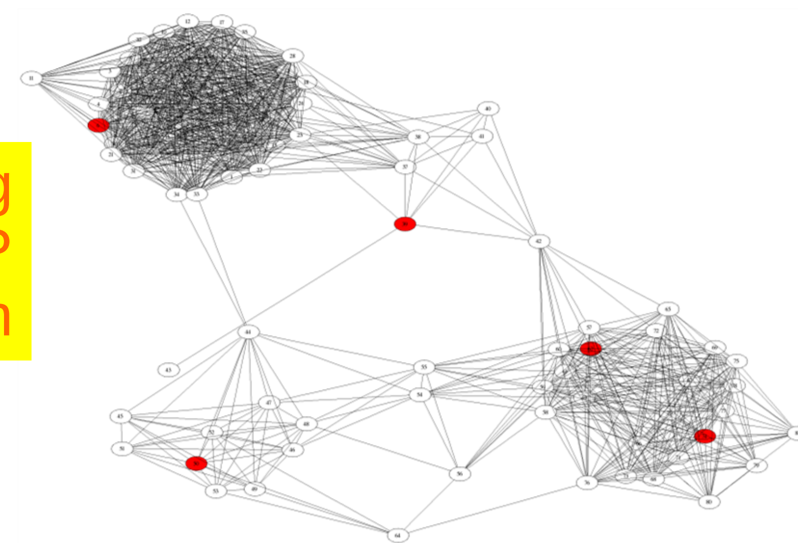
SPOT5 #509 (*Constraints* 4(3), 1999)

Mendelian
Error
Detection



langladeM7 sheep pedigree
(*Constraints* 13(1), 2008)

Tag
SNP
Selection



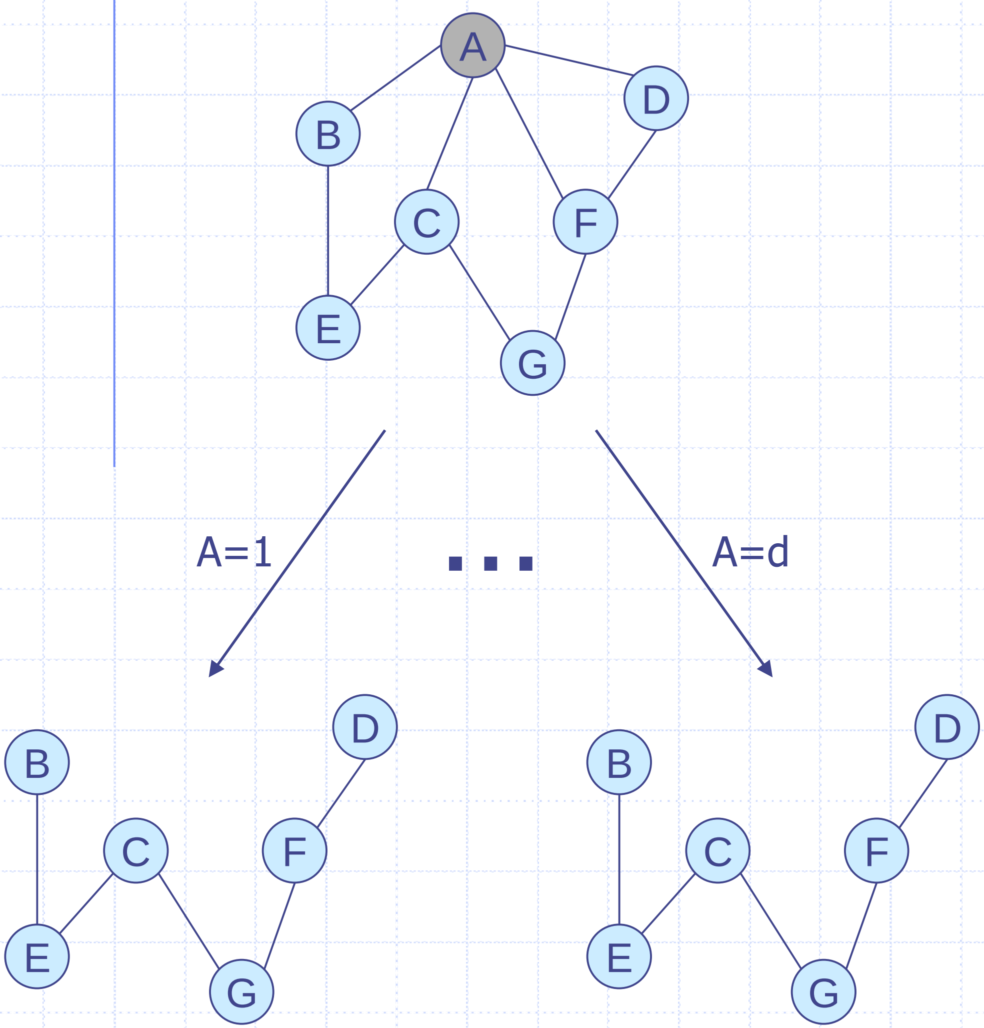
HapMap chr01 $r^2 \geq 0.8$ #14481
(*Bioinformatics* 22(2), 2006)

Search & Variable Elimination

- Condition, condition, condition … and then only eliminate (*Cycle-Cutset*)
- Eliminate, eliminate, eliminate … and then only search
- Interleave conditioning and elimination

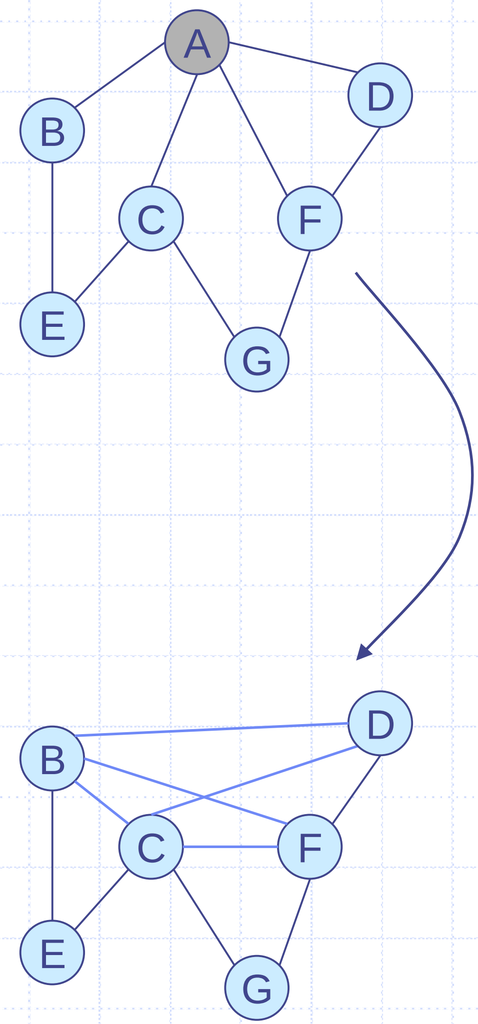
Conditioning vs. Elimination

Conditioning (search)



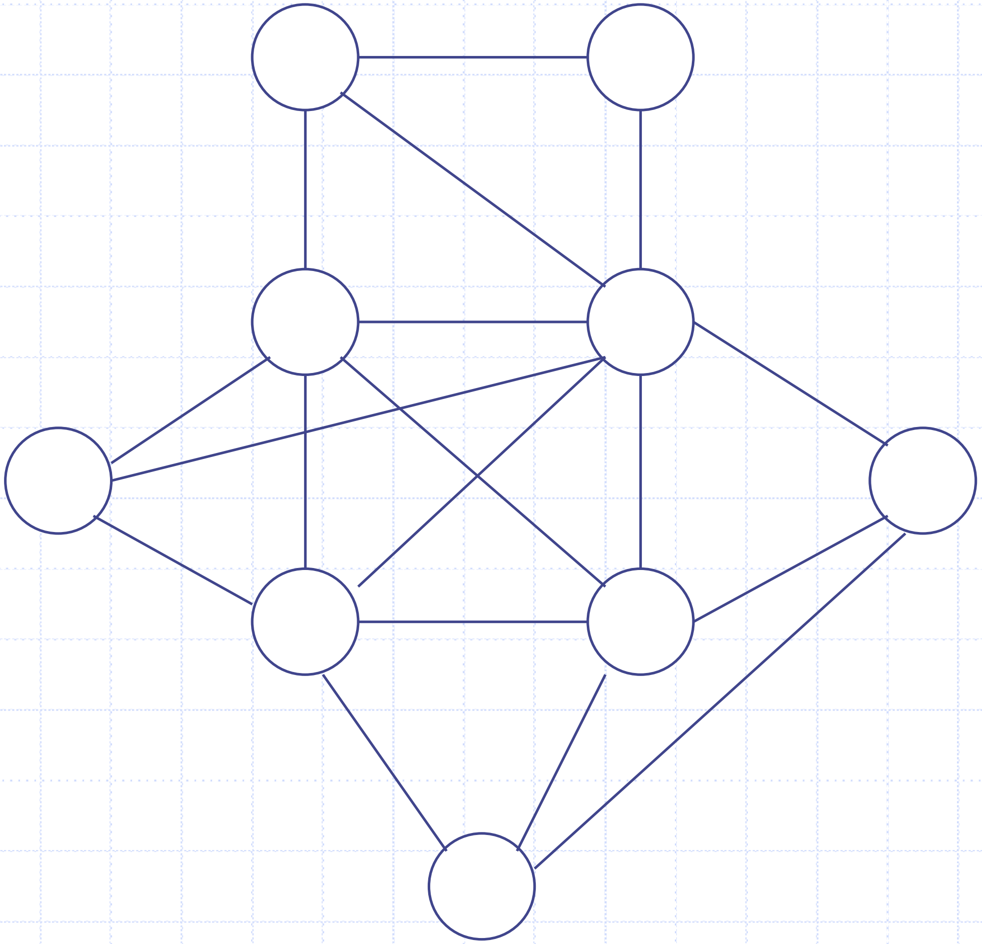
d “sparser” problems

Elimination (inference)

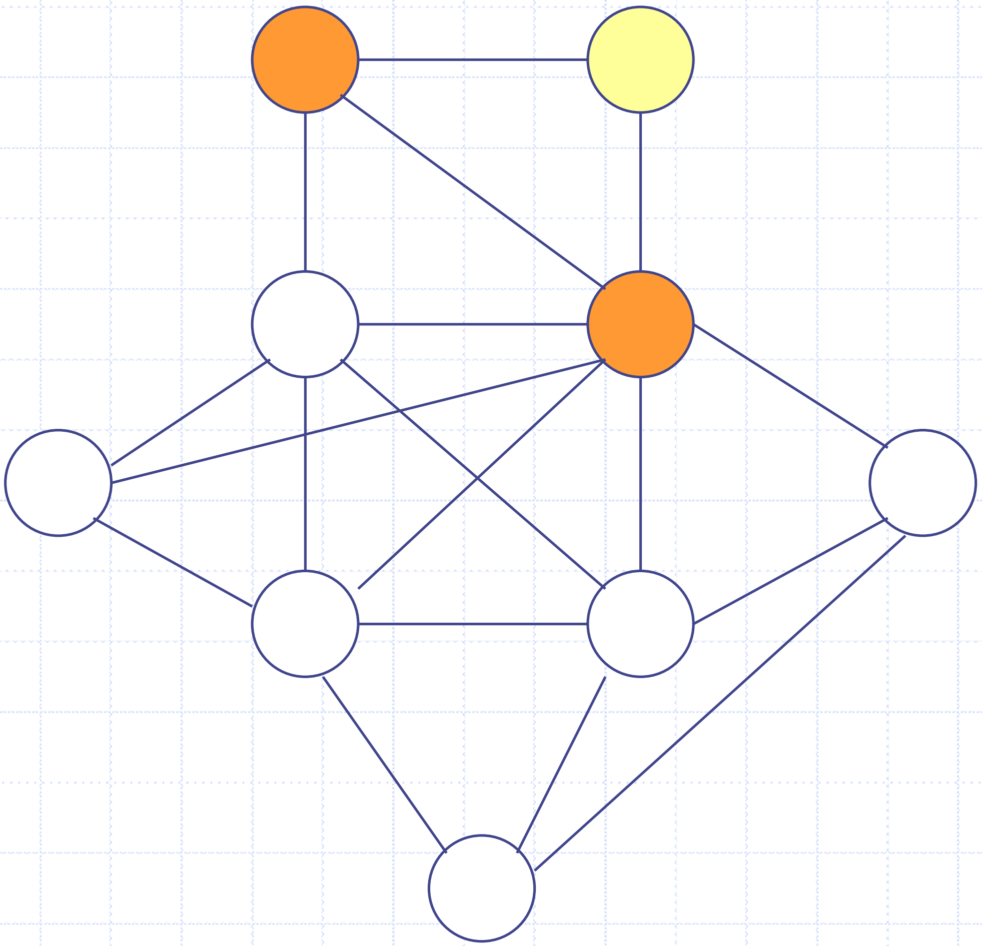


1 “denser” problem

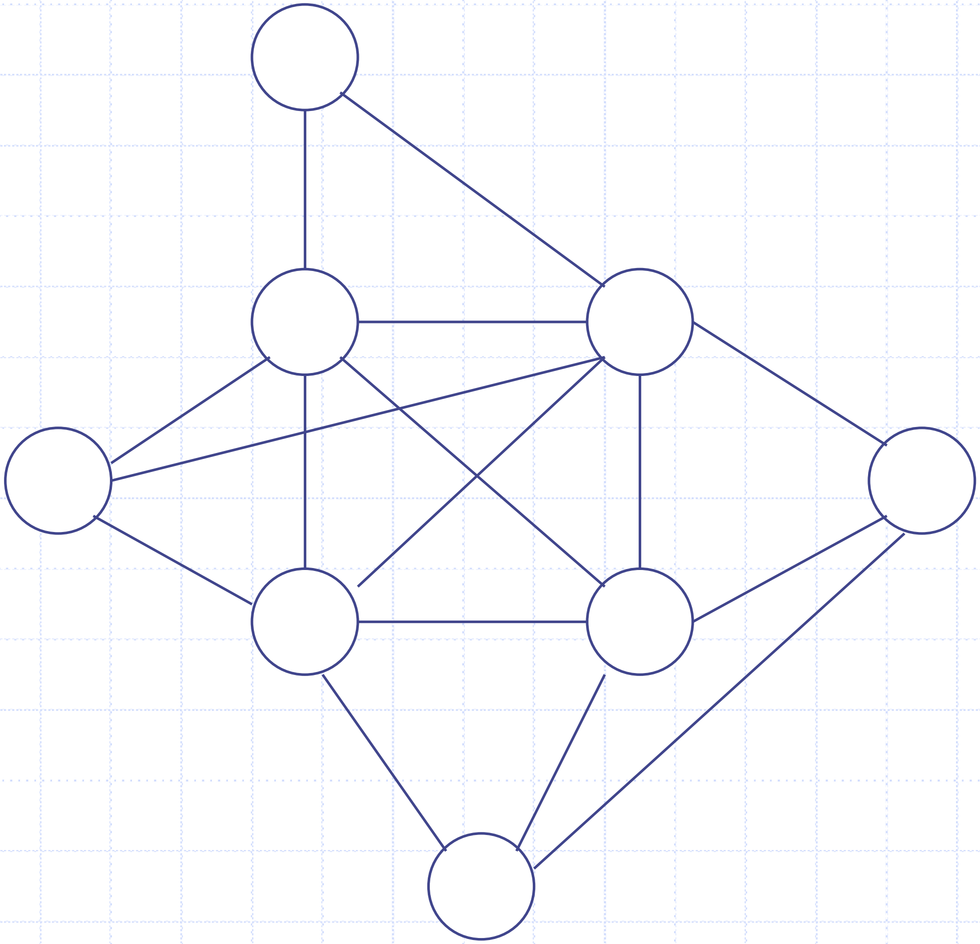
Interleaving Conditioning and Elimination BB-VE(2) (Larrosa & Dechter, CP 2002)



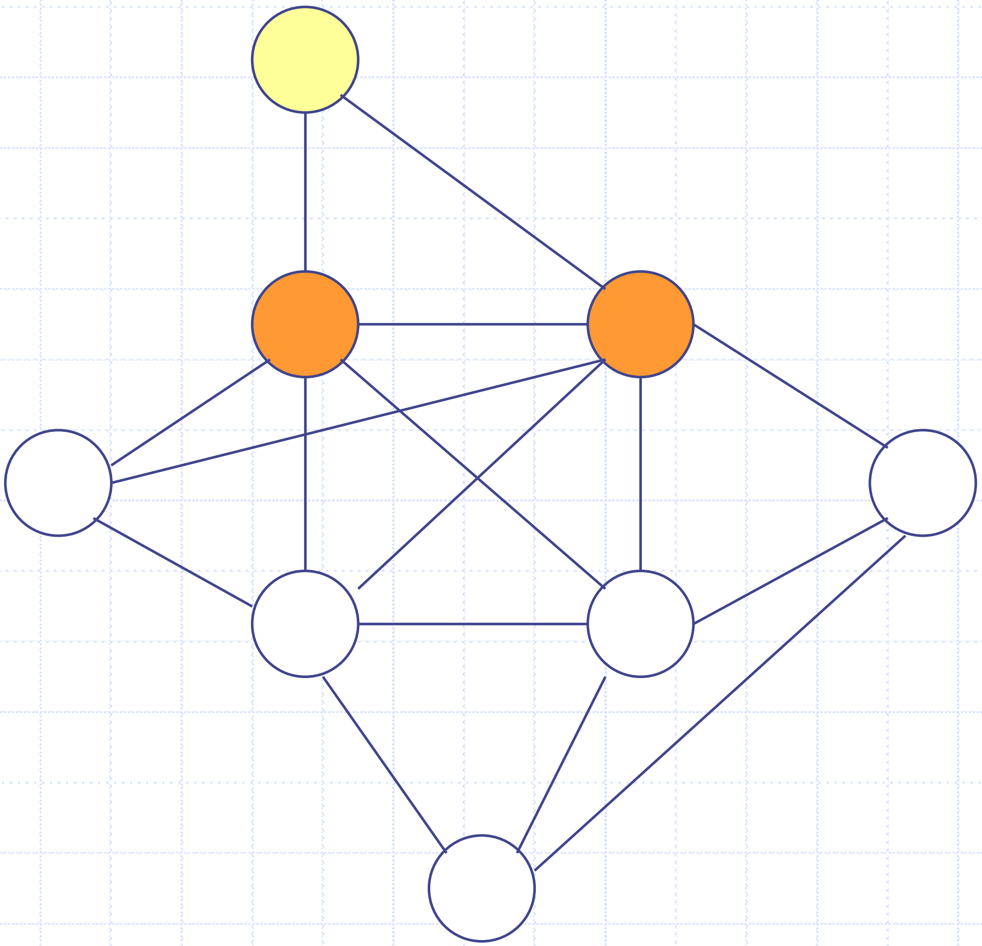
Interleaving Conditioning and Elimination BB-VE(2)



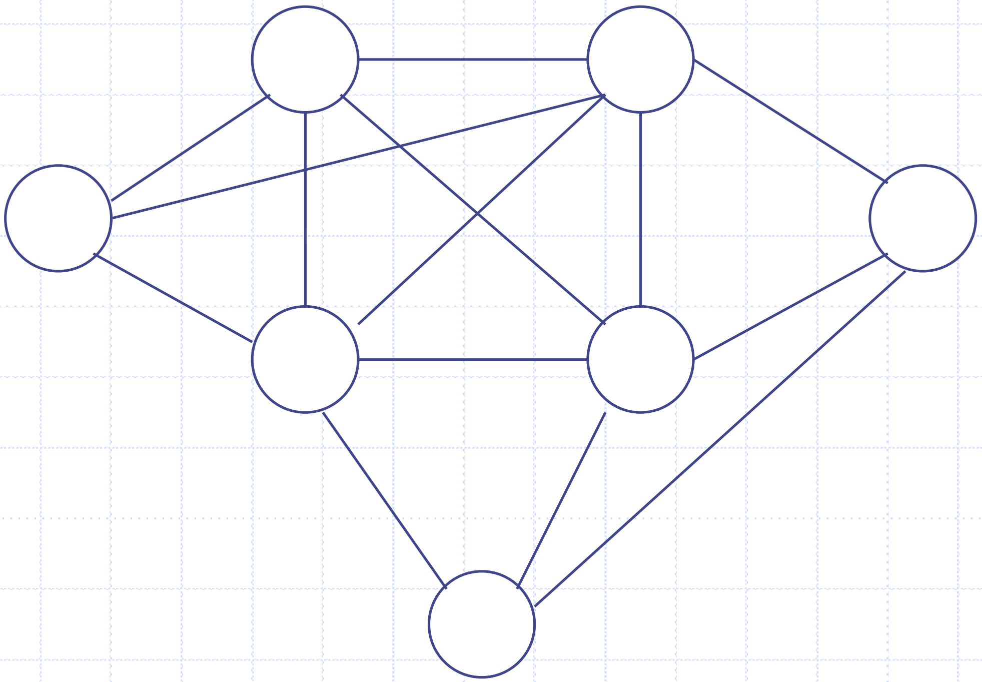
Interleaving Conditioning and Elimination BB-VE(2)



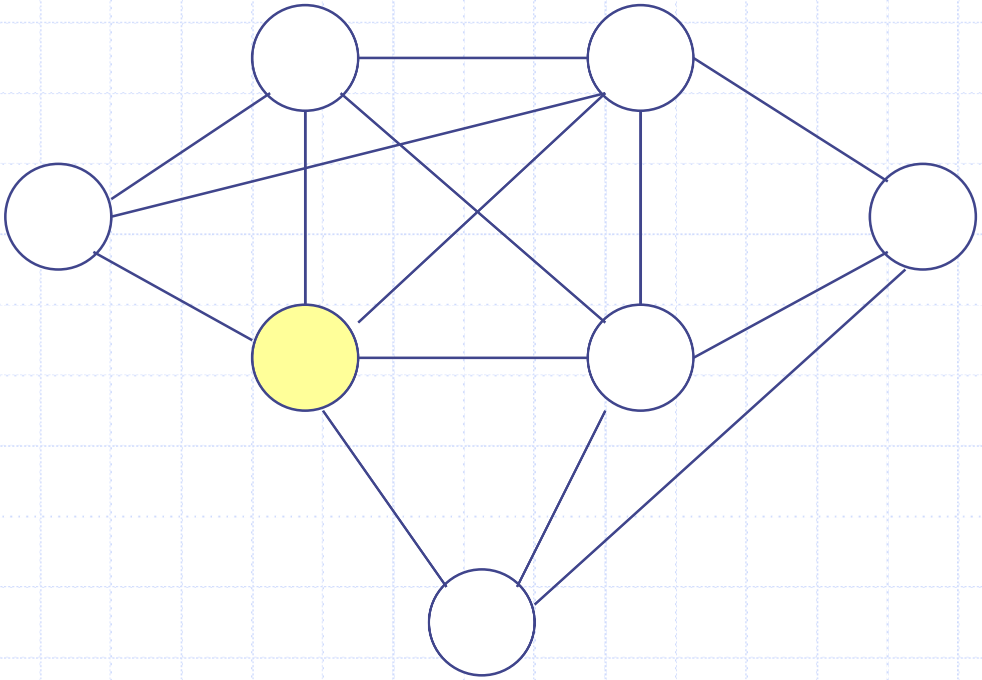
Interleaving Conditioning and Elimination BB-VE(2)



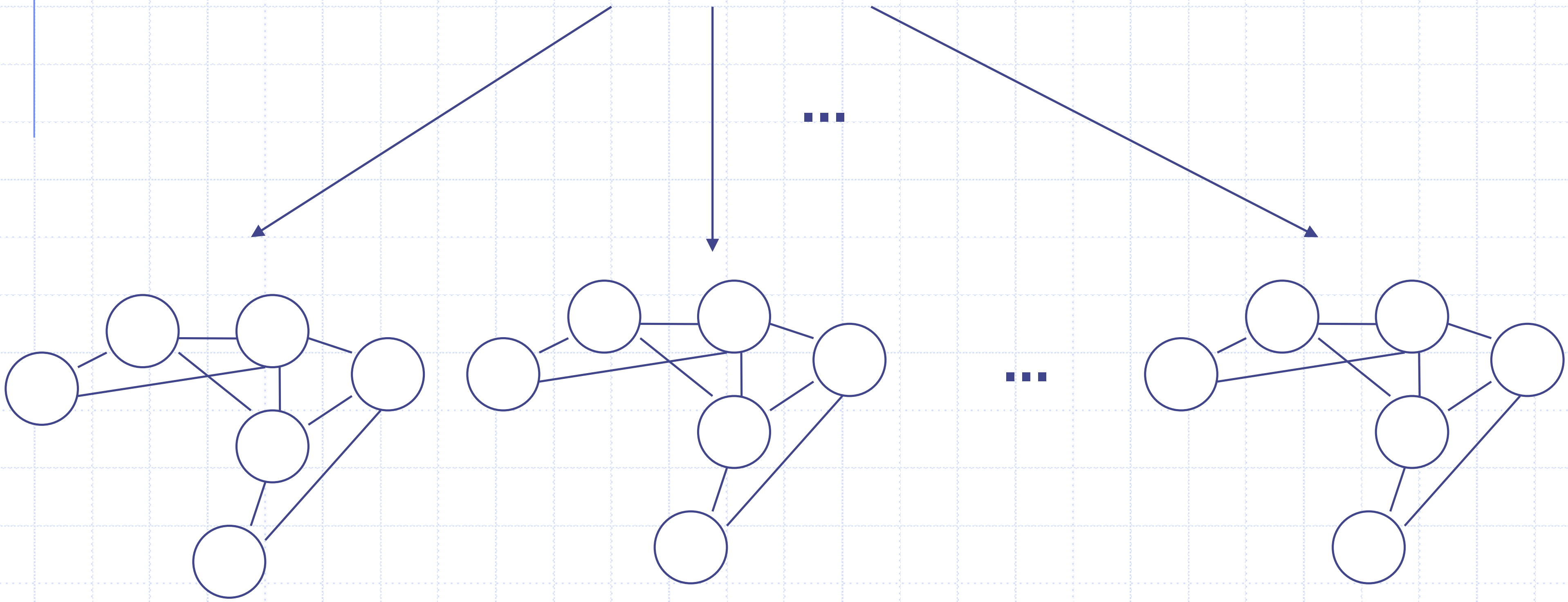
Interleaving Conditioning and Elimination BB-VE(2)



Interleaving Conditioning and Elimination BB-VE(2)

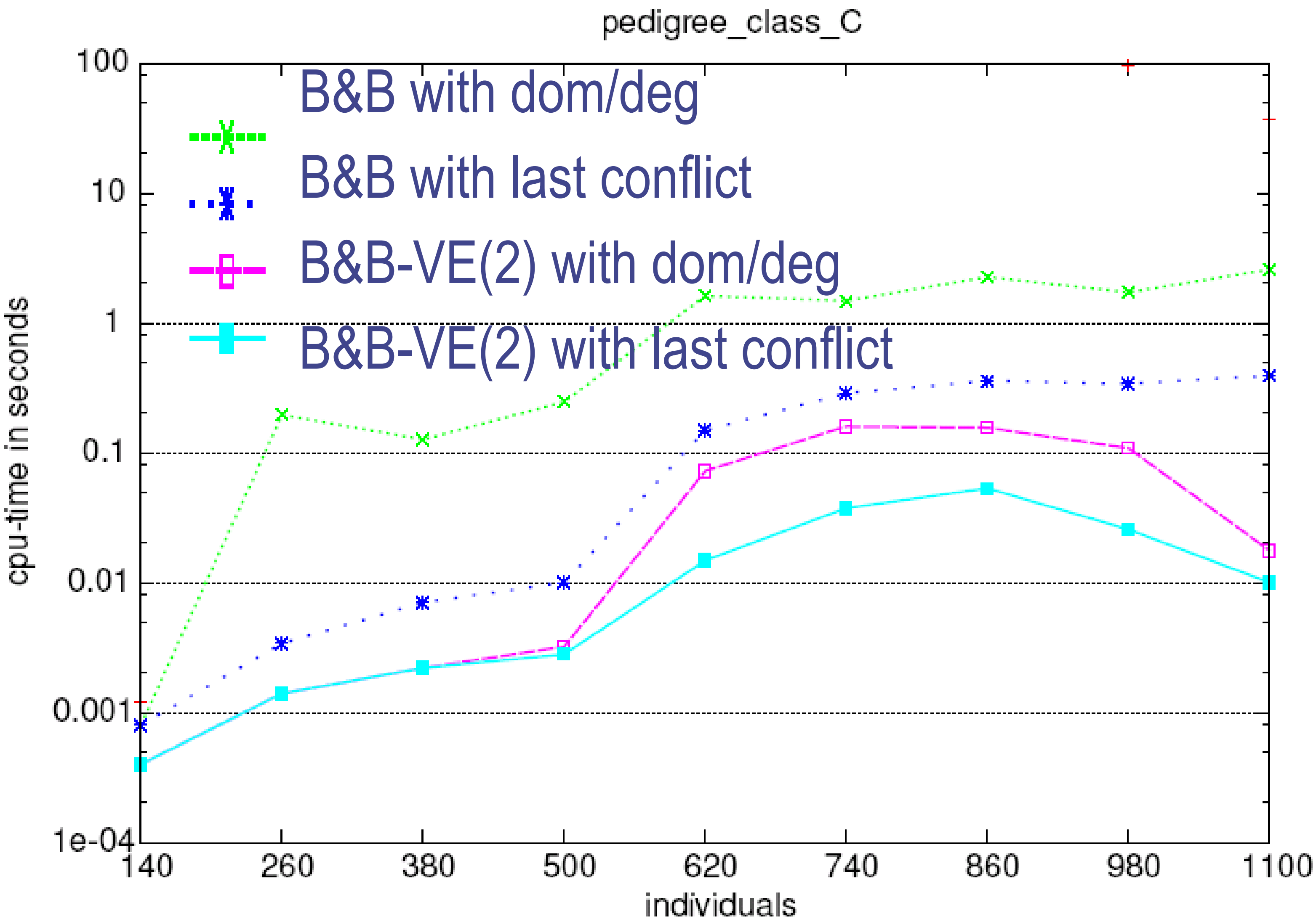


Interleaving Conditioning and Elimination BB-VE(2)



Pedigree

- toulbar2 v0.5 with EDAC3 and binary branching
- Minimize the number of genotypings to be removed
- CPU time in seconds to find and prove optimality on a linux PC 3 GHz with 16 GB



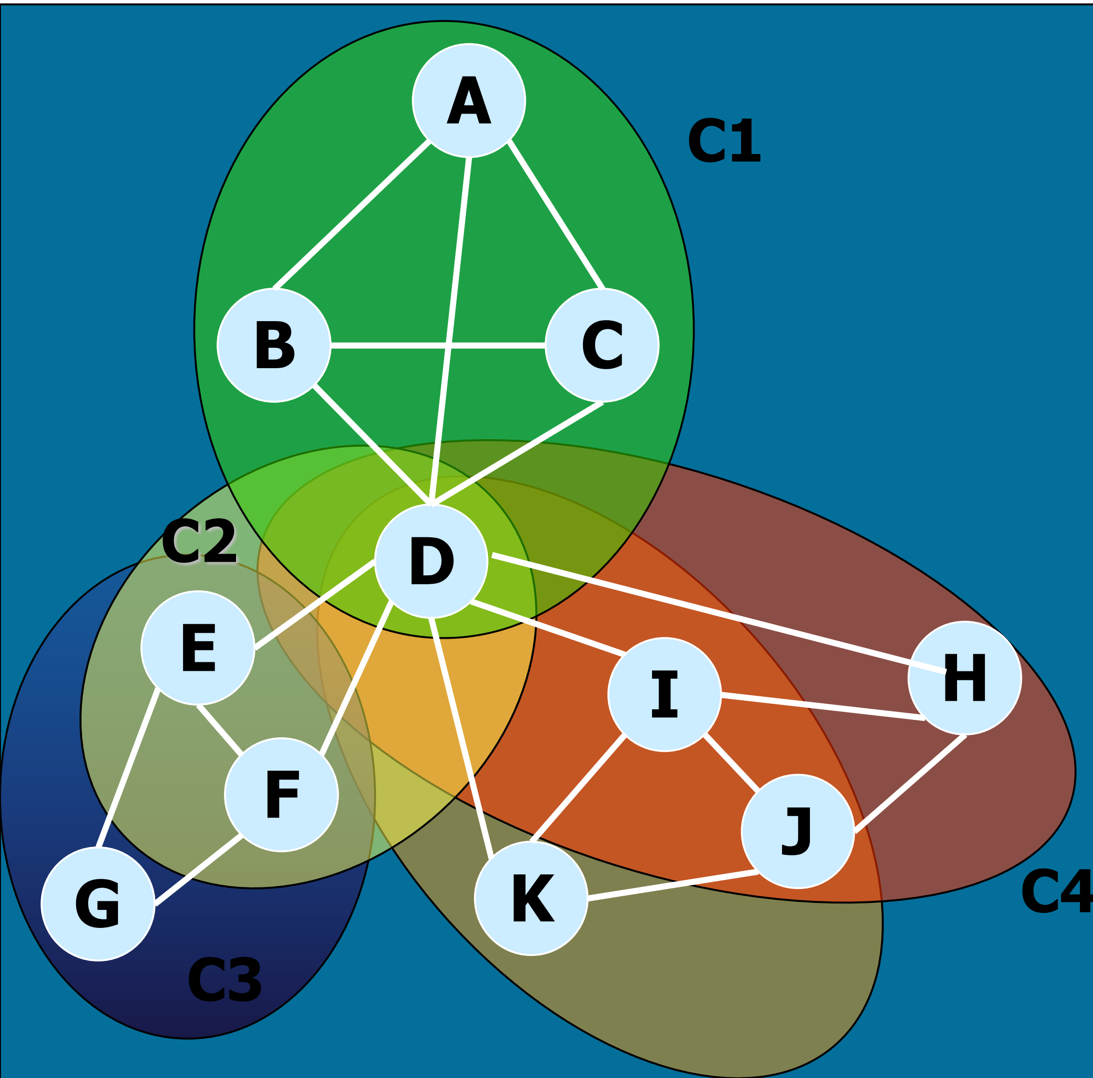
(Sanchez *et al*, Constraints 2008)

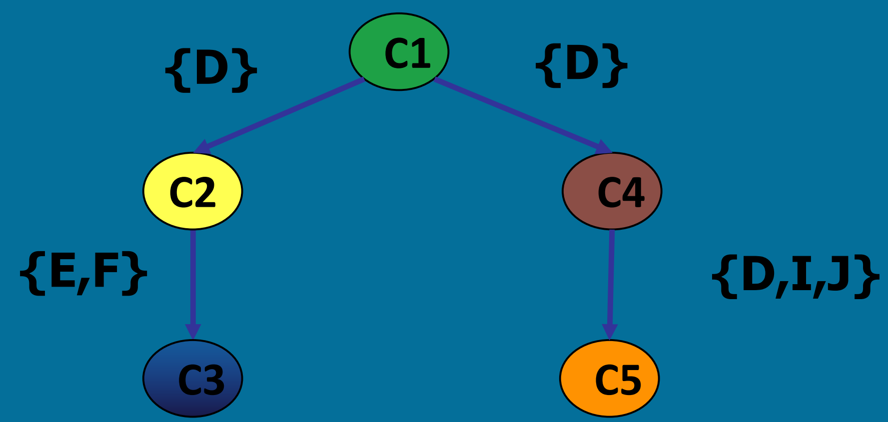
Search & Cluster Tree Elimination

- Depth-First Branch and Bound exploiting a tree decomposition with:
  - A restricted variable ordering
  - Graph-based backjumping
  - Graph-based learning

  $\Rightarrow$ Lazy elimination of subproblems using search

Tree Decomposition

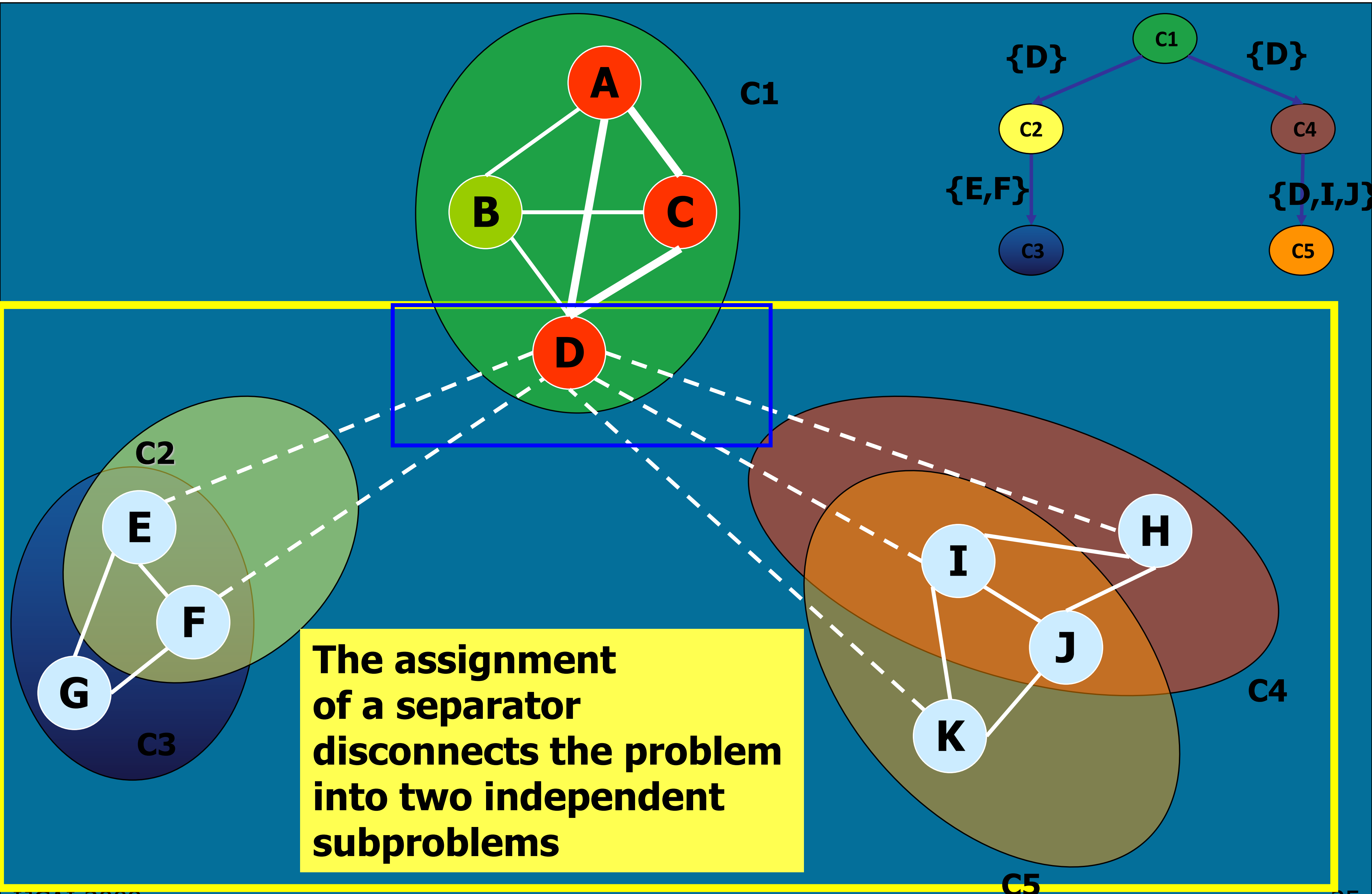




The set of clusters covers the set of variables and the set of cost functions

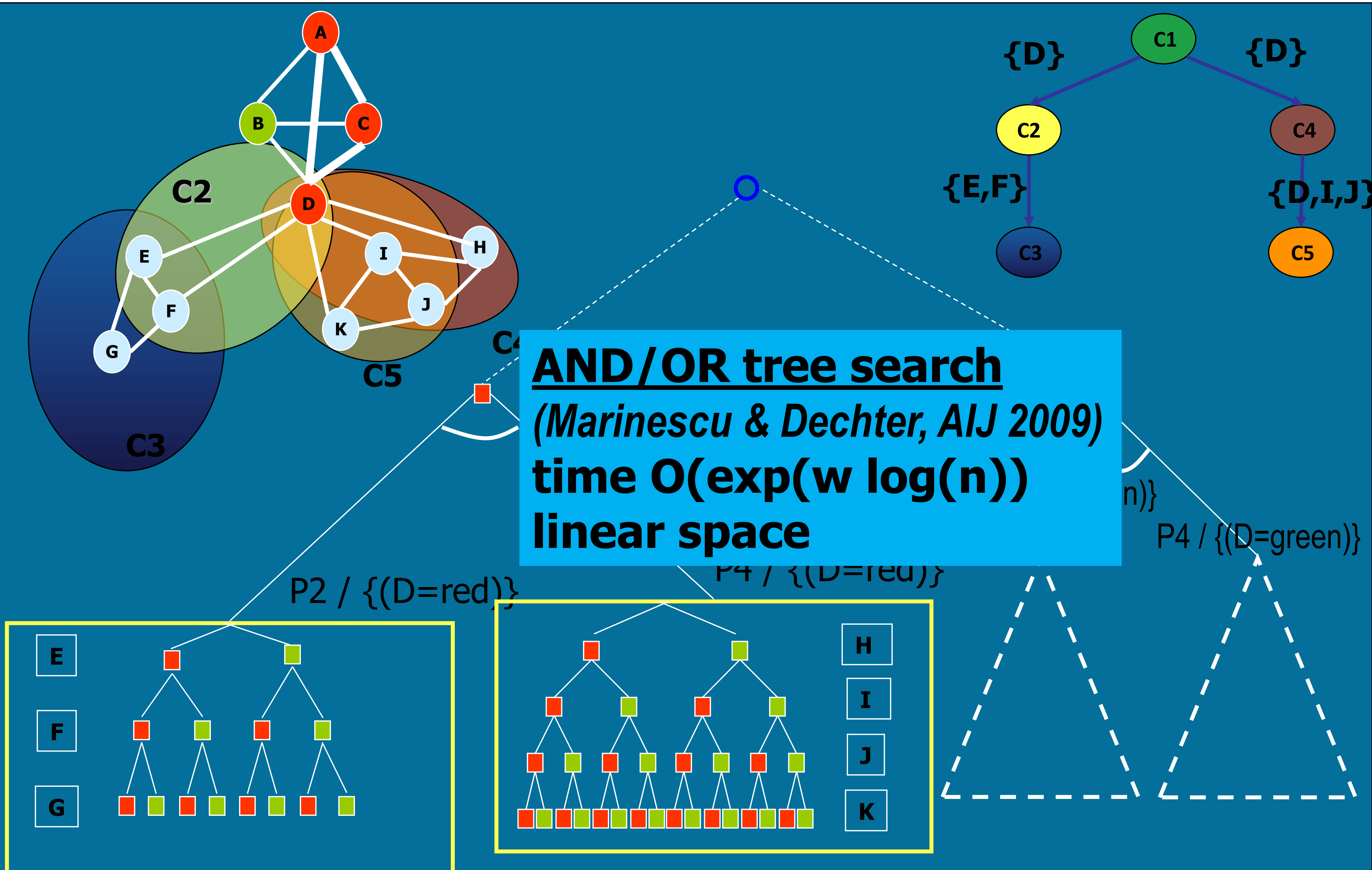
Separator = intersection between two connected clusters

Search with Tree Decomposition



The assignment of a separator disconnects the problem into two independent subproblems

Search with Tree Decomposition



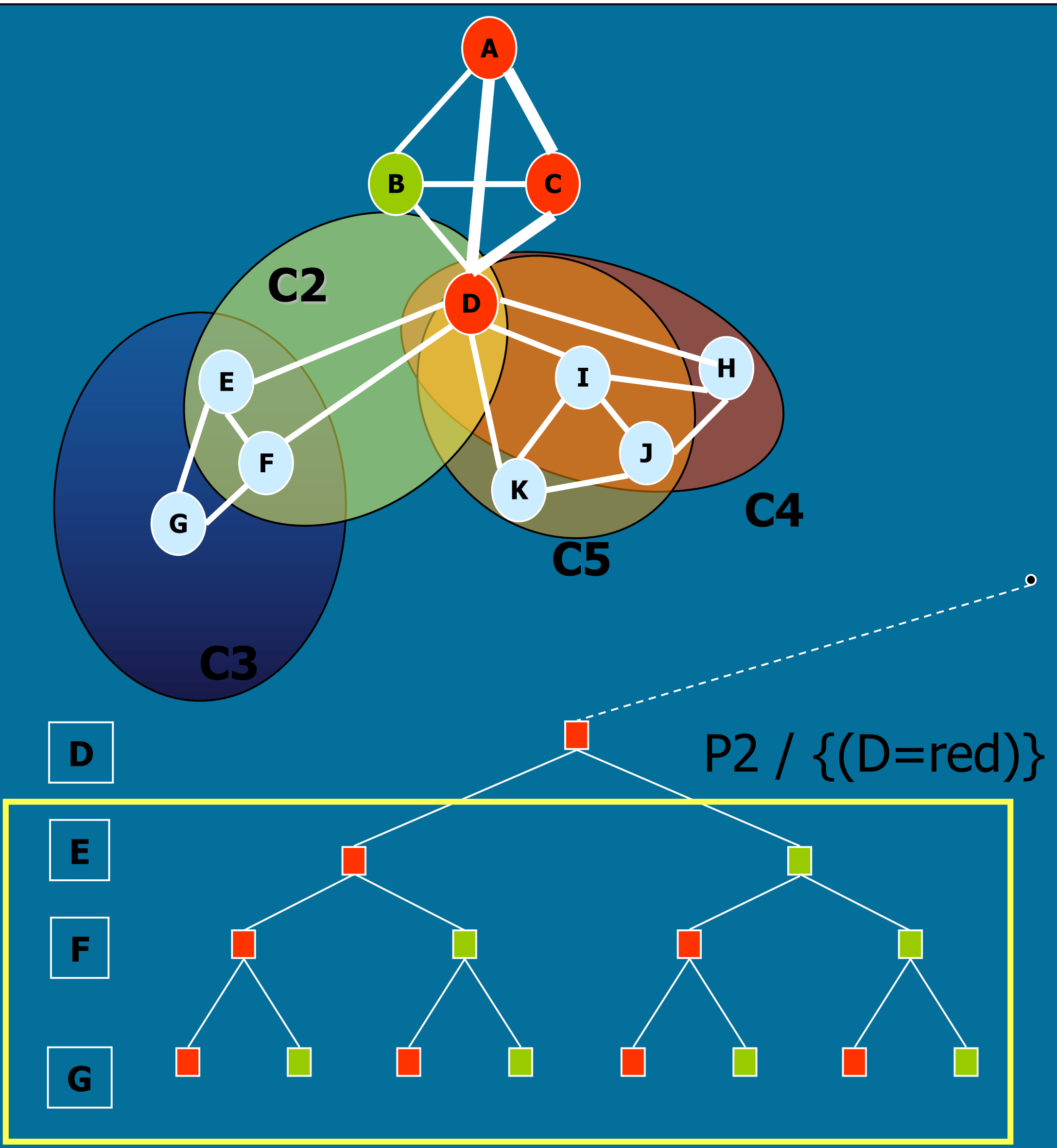
AND/OR tree search

(Marinescu & Dechter, AIJ 2009)

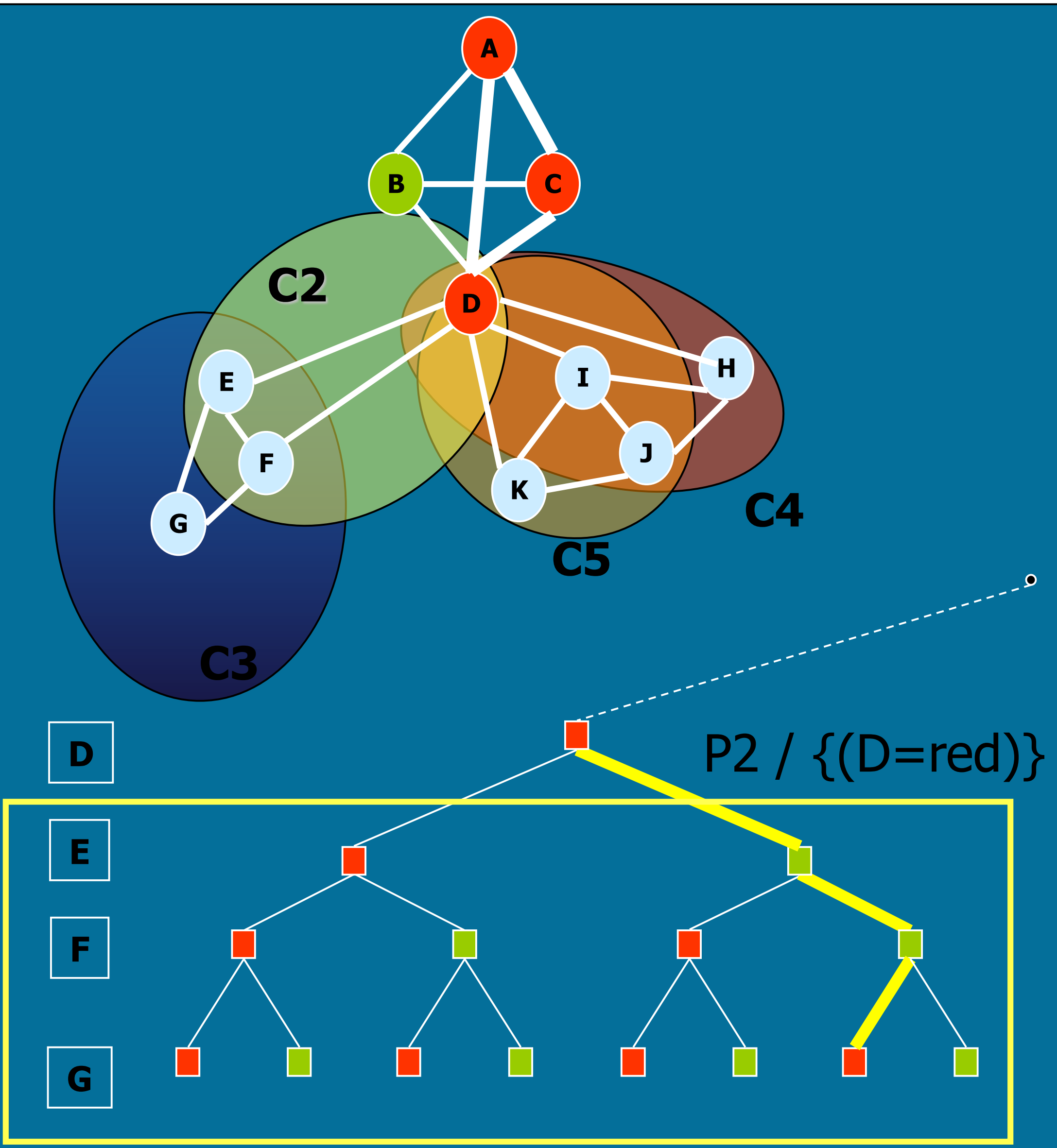
time O(exp(w log(n))

linear space

Search with Tree Decomposition



Search with Tree Decomposition



Record the optimum of P2 / {(D=red)}

AND/OR graph search

(Marinescu & Dechter, AIJ 2009)

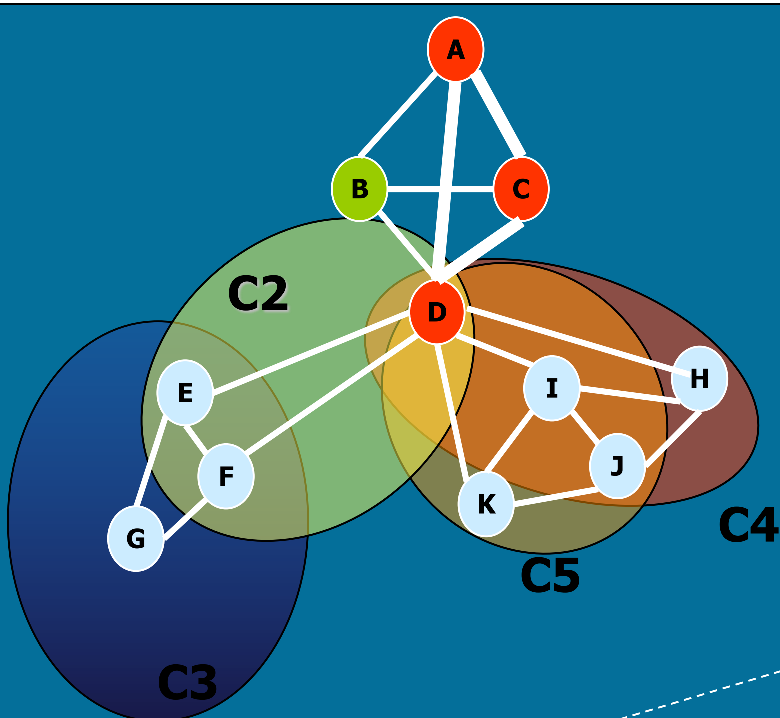
time O(exp(w))

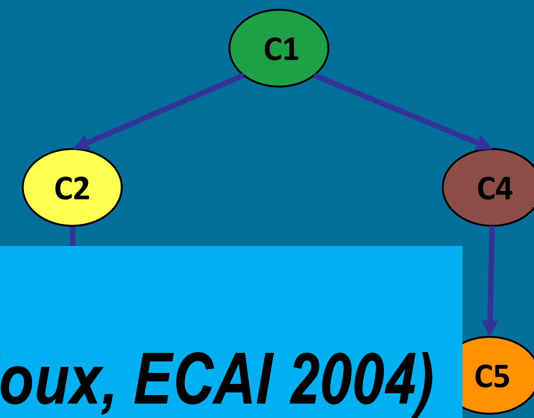
space O(exp(w))

bound k = 5.

It may be useless to compute the optimum of P2 / {(D=red)}, only a lower bound is needed!

Backtrack bounded by Tree Decomposition





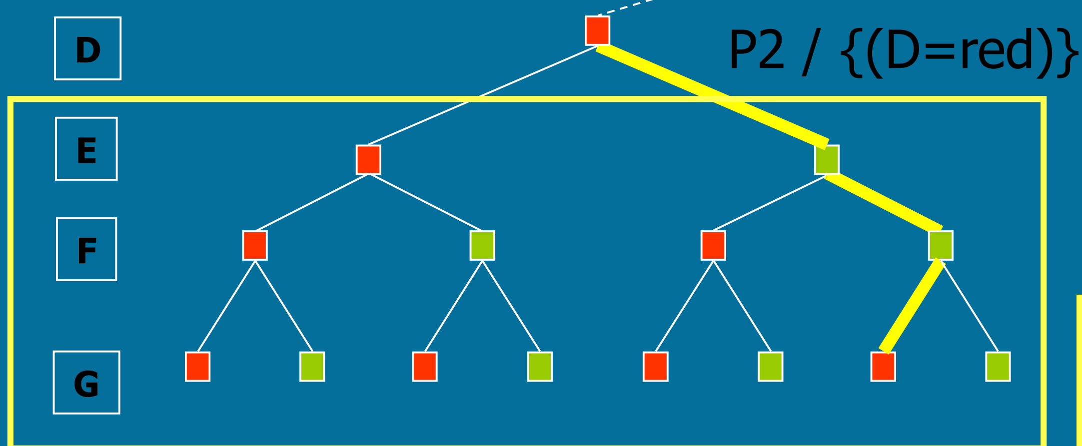
BTD

(Jégou & Terrioux, ECAI 2004)

(de Givry et al., AAAI 2006)

time O(k*exp(w))

space O(exp(w))



It may be useless to compute the optimum of P2, only a lower bound is needed!

Add a local upper bound:

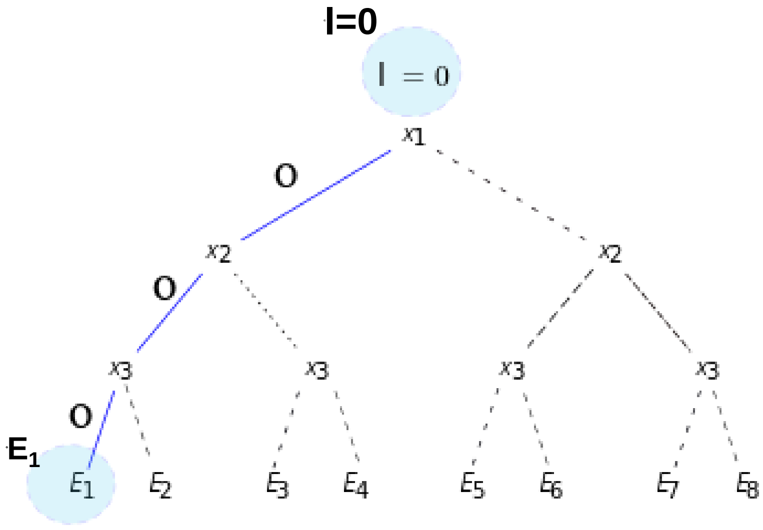
$UB_{P2\,/\,\{(D=red)\}} = k - 3 - LB_{P4\,/\,\{(D=red)\}}$

$$UB_{P2\,/\,\{(D=red)\}} = k - 3 - \max\Big(\underbrace{f_{\varnothing}^{C4} + f_{\varnothing}^{C5}}_{\text{Maintaining local consistency}},\ \underbrace{LB_{P4\,/\,\{(D=red)\}}}_{\text{Recorded during search}}\Big)$$

Bibliography

- For hybrids of search and inference, see the chapter 10 in *Constraint Processing*, Dechter, Morgan Kaufmann, 2003.
- For exploiting tree decomposition, see
  - "*Exploiting Tree Decomposition and Soft Local Consistency in Weighted CSP*", de Givry, Schiex & Verfaillie , AAAI 2006.
  - "*Memory intensive AND/OR search for combinatorial optimization in graphical models (Part I&II)*", Marinescu & Dechter, AIJ 2009.

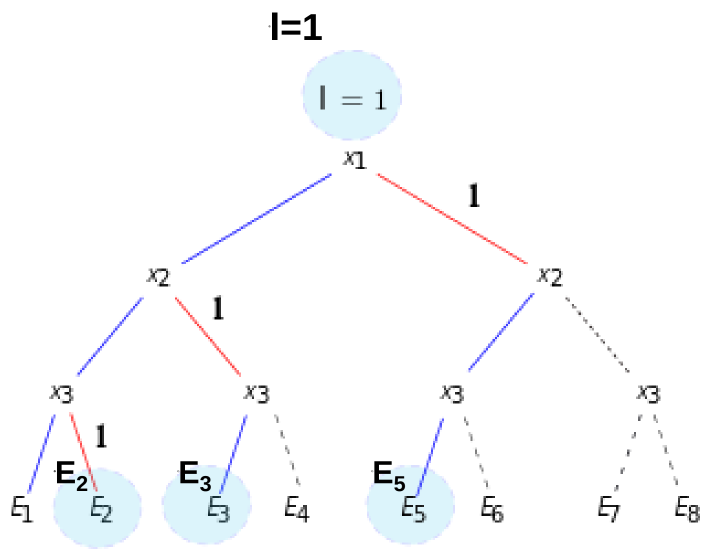
Limited Discrepancy Search *(Ginsberg 95)*



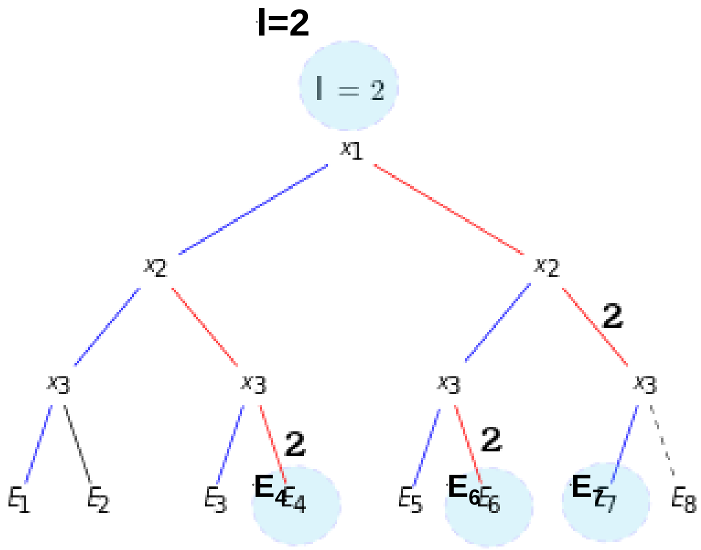
- Small example with 3 variables and 2 values per domain

Limited Discrepancy Search

- **Small example with 3 variables and 2 values per domain**



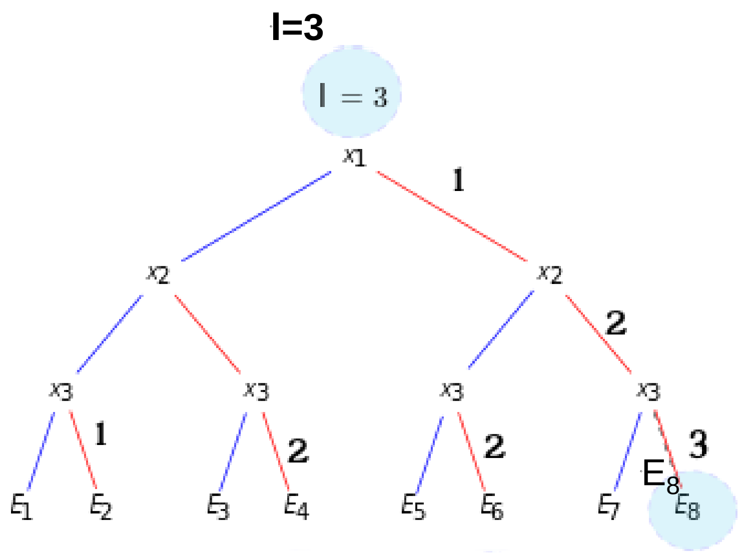
Limited Discrepancy Search *(Ginsberg 95)*



Limited Discrepancy Search *(Ginsberg 95)*

$l_{max} = n * (d - 1)$: in this case, $l_{max}=3*(2-1)=3$

Full exploration



l=3 ⇒ optimality proof

In practice, it occurs before l_{max} thanks to bounding and pruning

Bibliography

- For LDS, see:
  Limited discrepancy search, Harvey and Ginsberg, *IJCAI 1995*.

- For partial tree search, see:

Nonsystematic backtracking search, Harvey, *PhD 1995*.

A unified framework for partial and hybrid search methods in constraint programming, Givry and Jeannin, *C&OR 2006*.

INCOP local search *IDWalk*

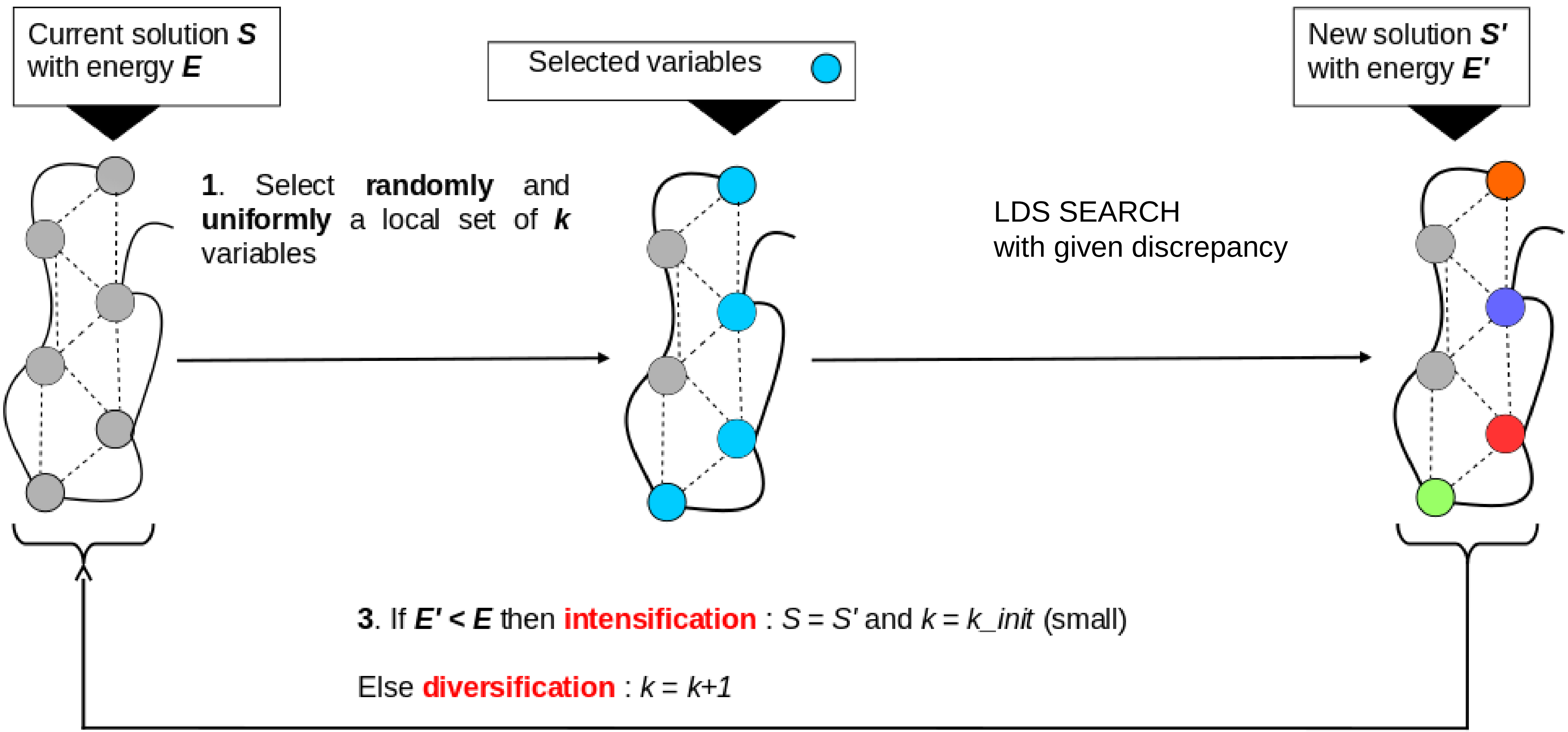
(Neveu *et al*, CP 2004)

- *IDWalk* performs *S* moves and returns the best solution found during the walk.
- A move examines at most *Max* candidate neighbors at random (flips among variables in conflicts):
  - If the cost of a neighbor is less than or equal to the cost of the current solution, then it is selected (intensification)
  - If no neighbors are selected, then chose one at random (diversification)

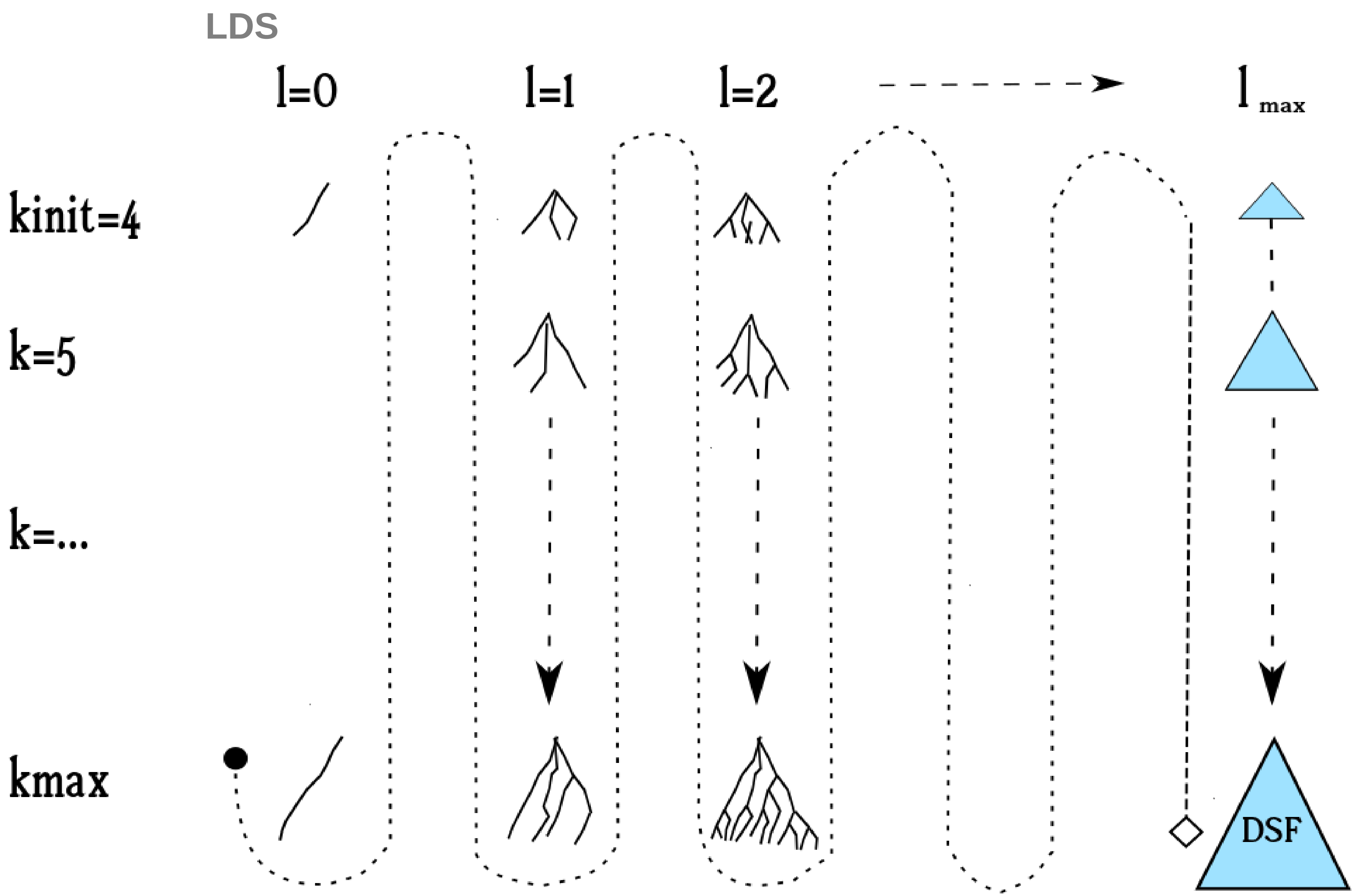
ID Walk: a Candidate List Strategy with a Simple Diversification Device.
B. Neveu, G. Trombettoni, F. Glover. LNCS 3258, Springer, p. 423--437, CP 2004

S= 100,000 ; *Max*=200 ; 3 repeats

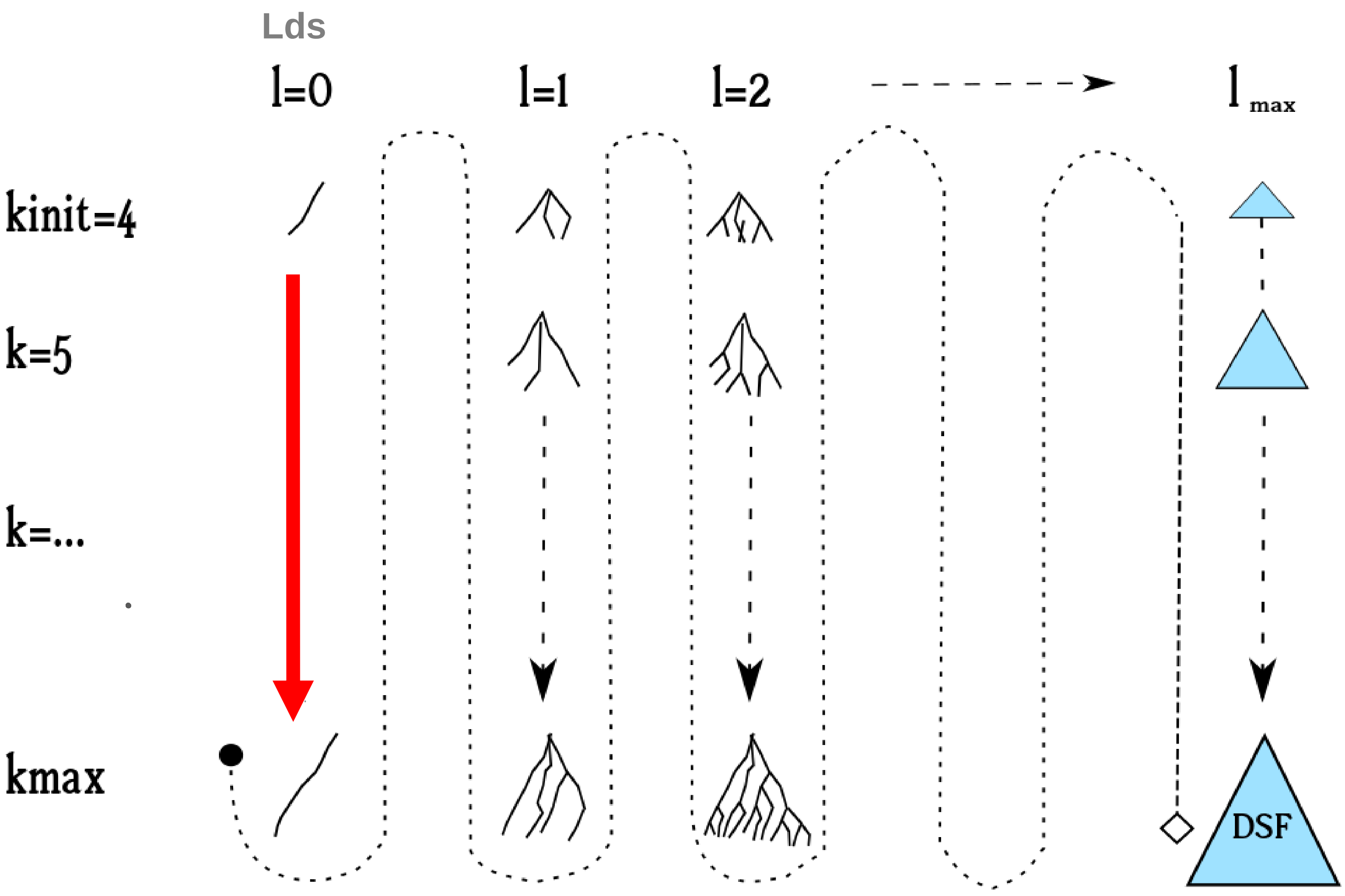
Variable Neighborhood Search *(Hansen 97)*



UDGVNS : Exploration of both k and l dimensions

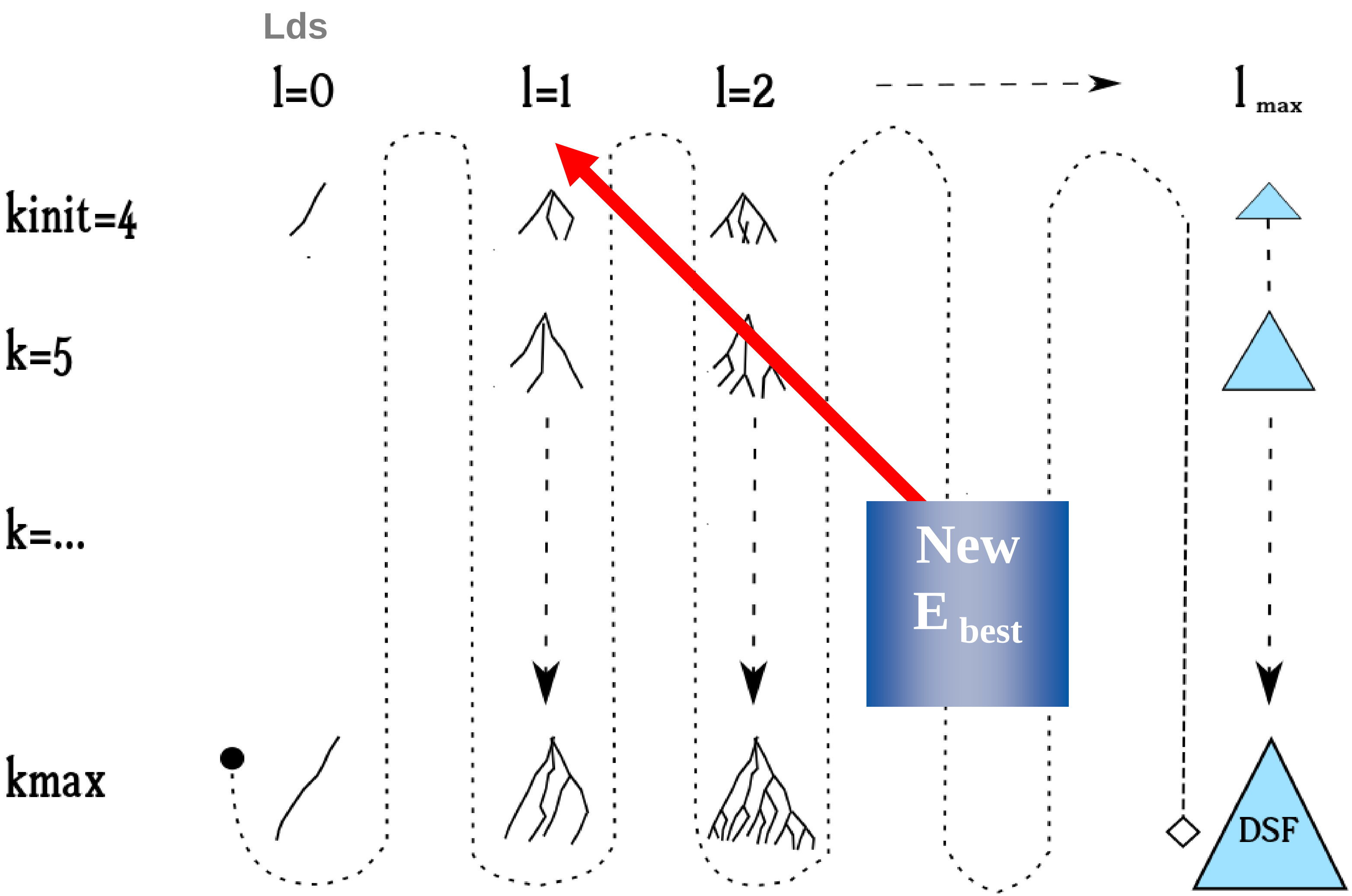


Step 1 : Initial solution

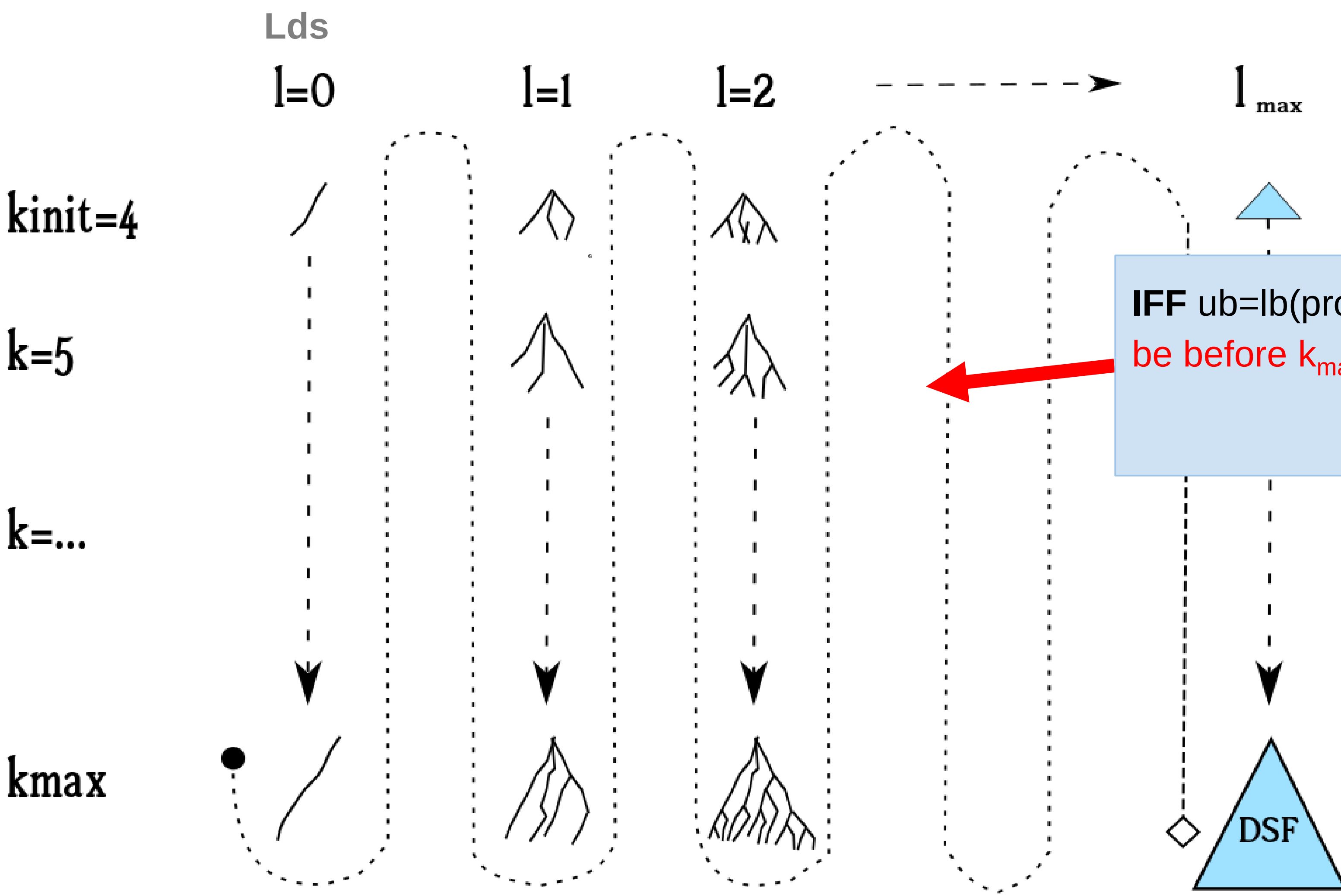


Greedy assignment

NEW SOLUTION WITH BETTER E → RESTART

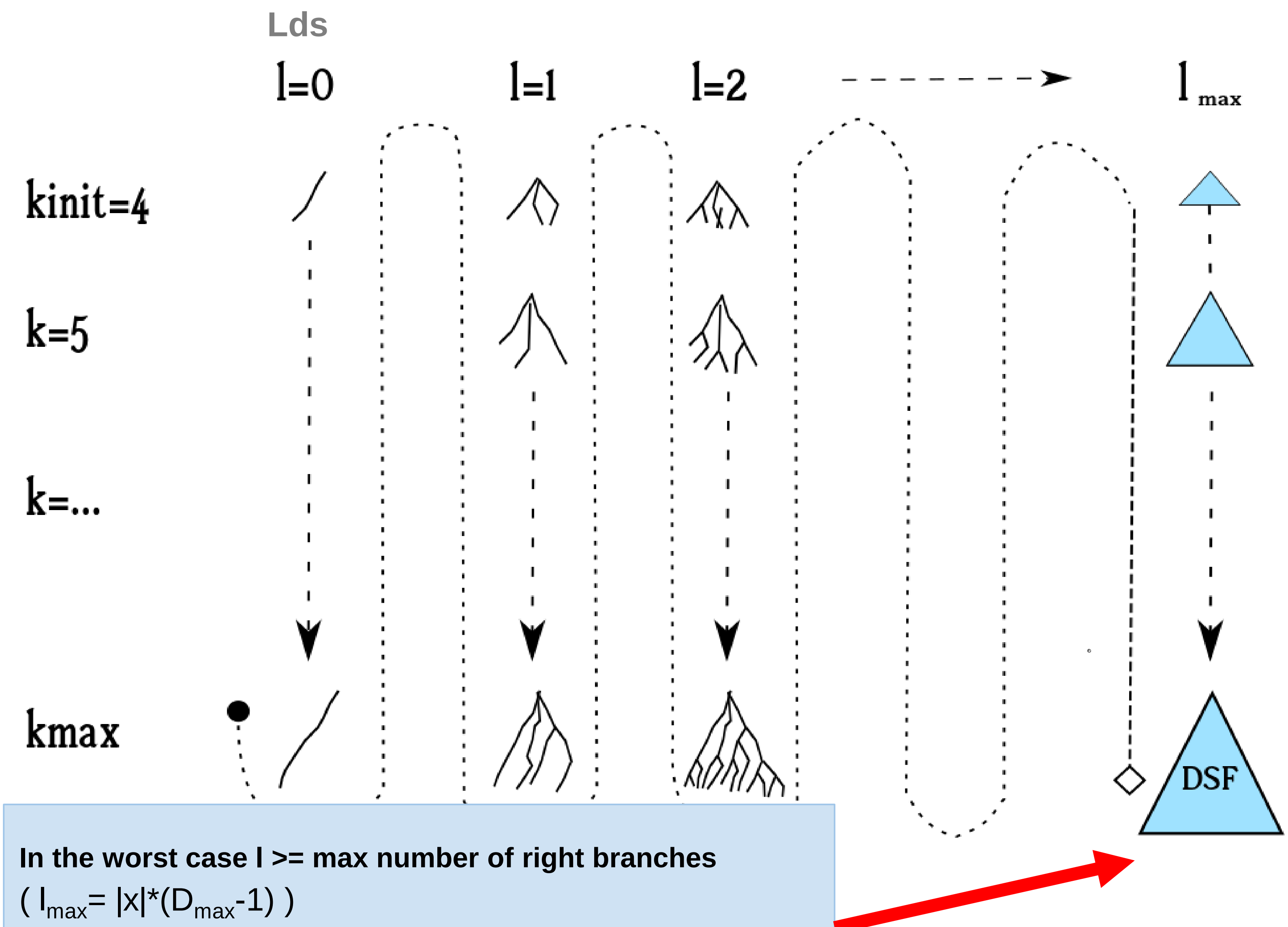


Proof of Optimality



IFF ub=lb(problem) can be before k_{max}

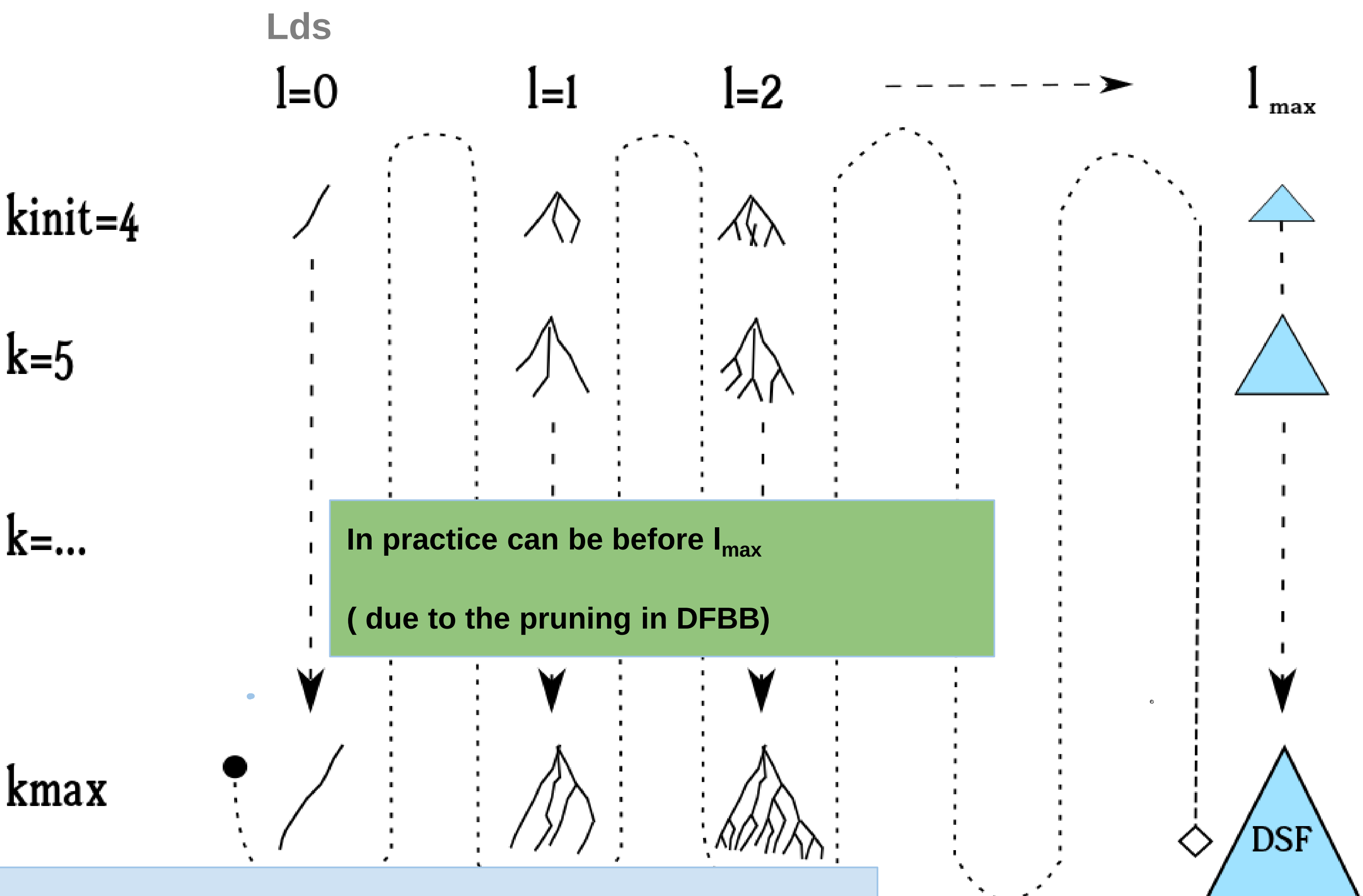
Proof of Optimality



In the worst case l >= max number of right branches
(l_{max}= |x|*(D_{max}-1))

Iff **k** = k_{max} = problem size

Proof of Optimality



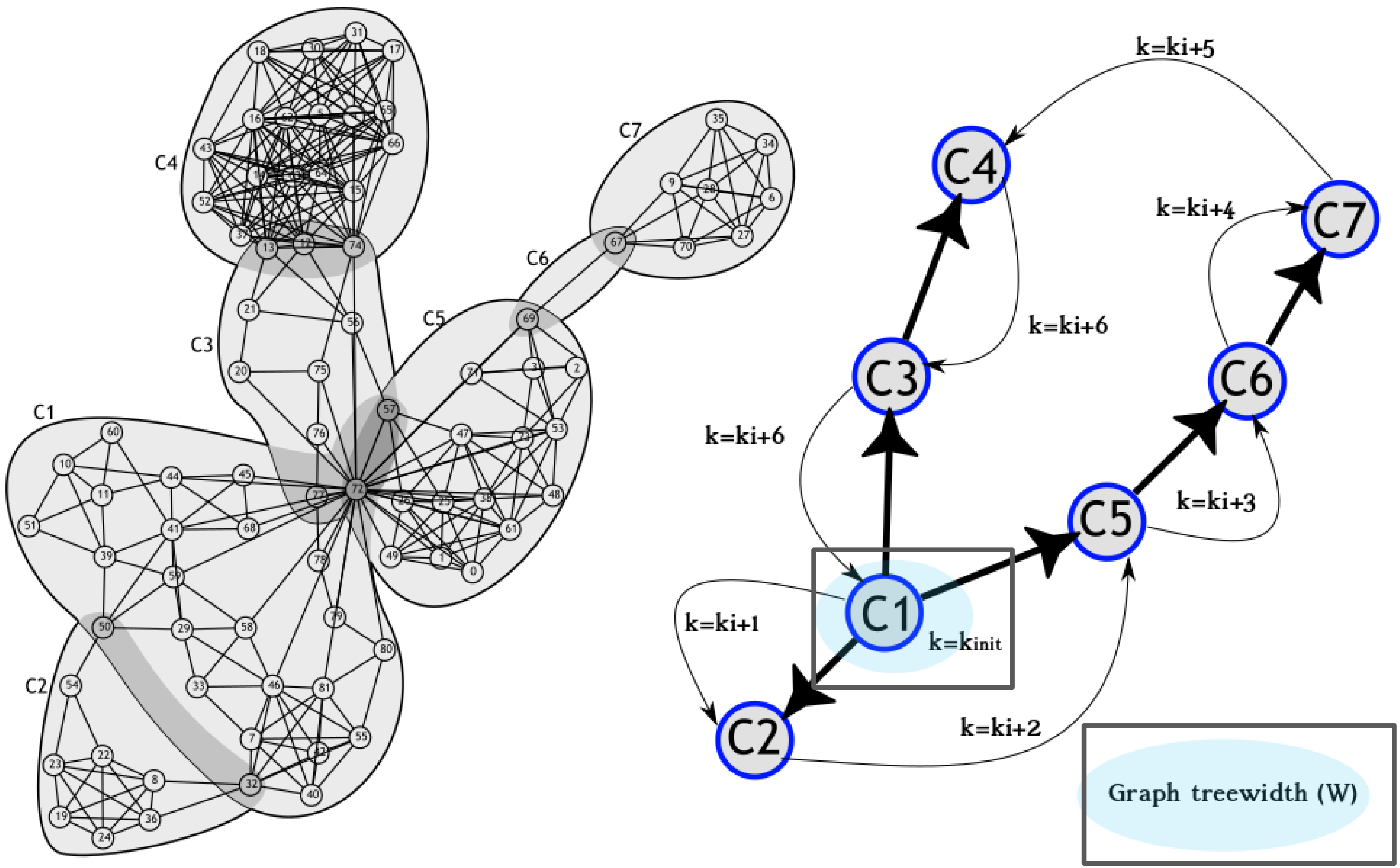
In practice can be before l_{max}

(due to the pruning in DFBB)

In the worst case l >= max number of right branches
(l_{max}= |x|*(D_{max}-1))

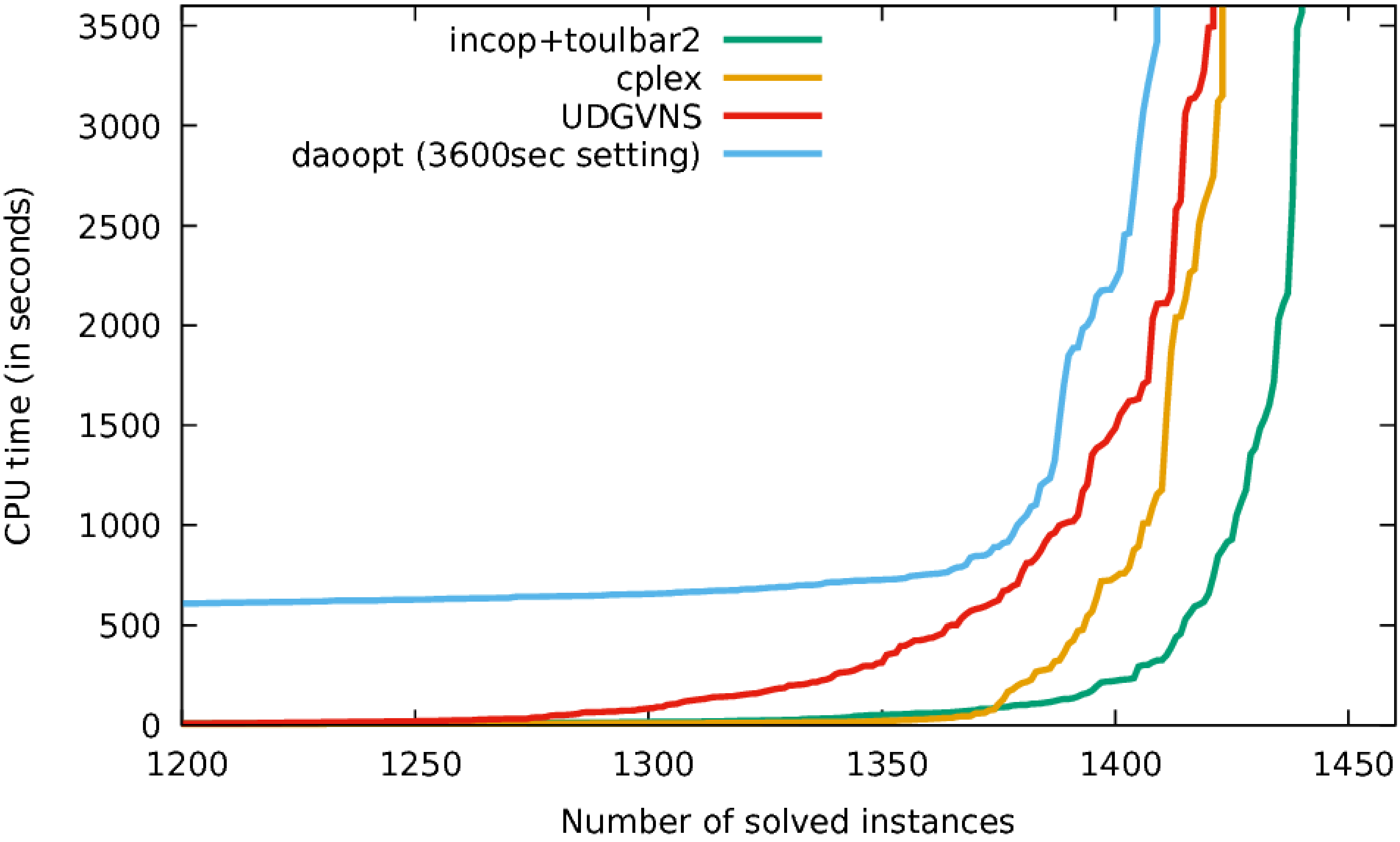
Iff **k** = k_{max} = problem size

Cluster visit in a topological order :



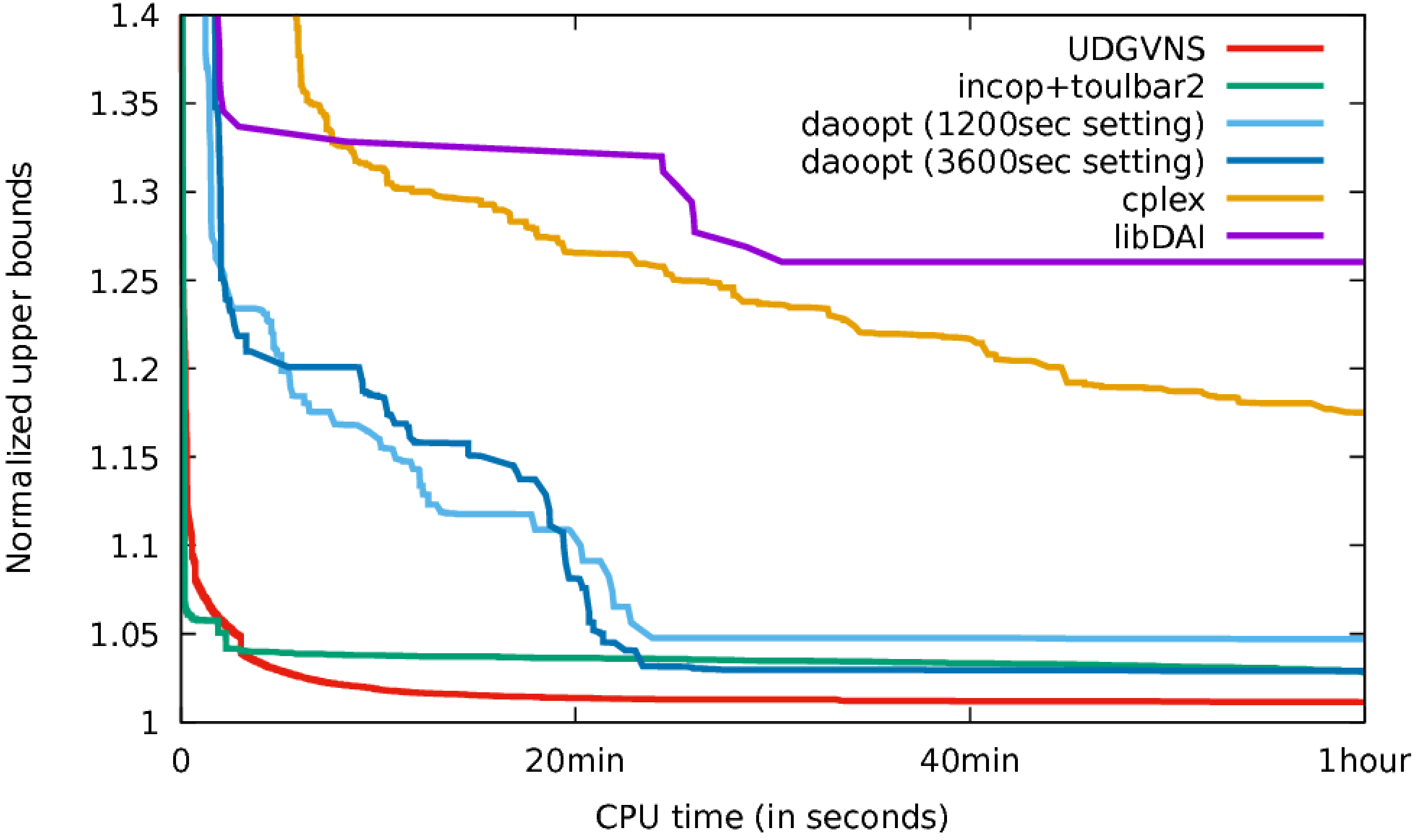
Results

(UAI17)



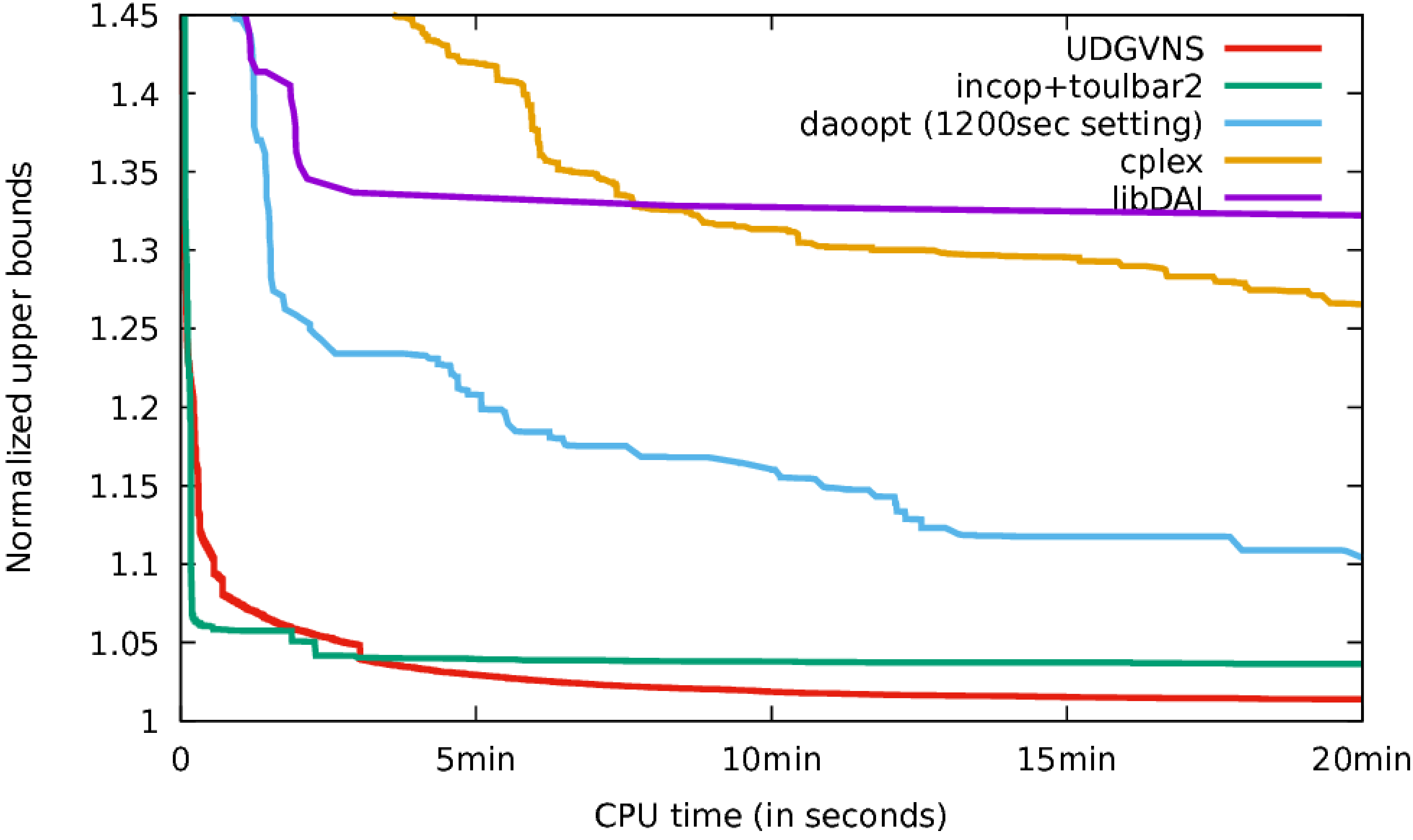
Results

(UAI17)

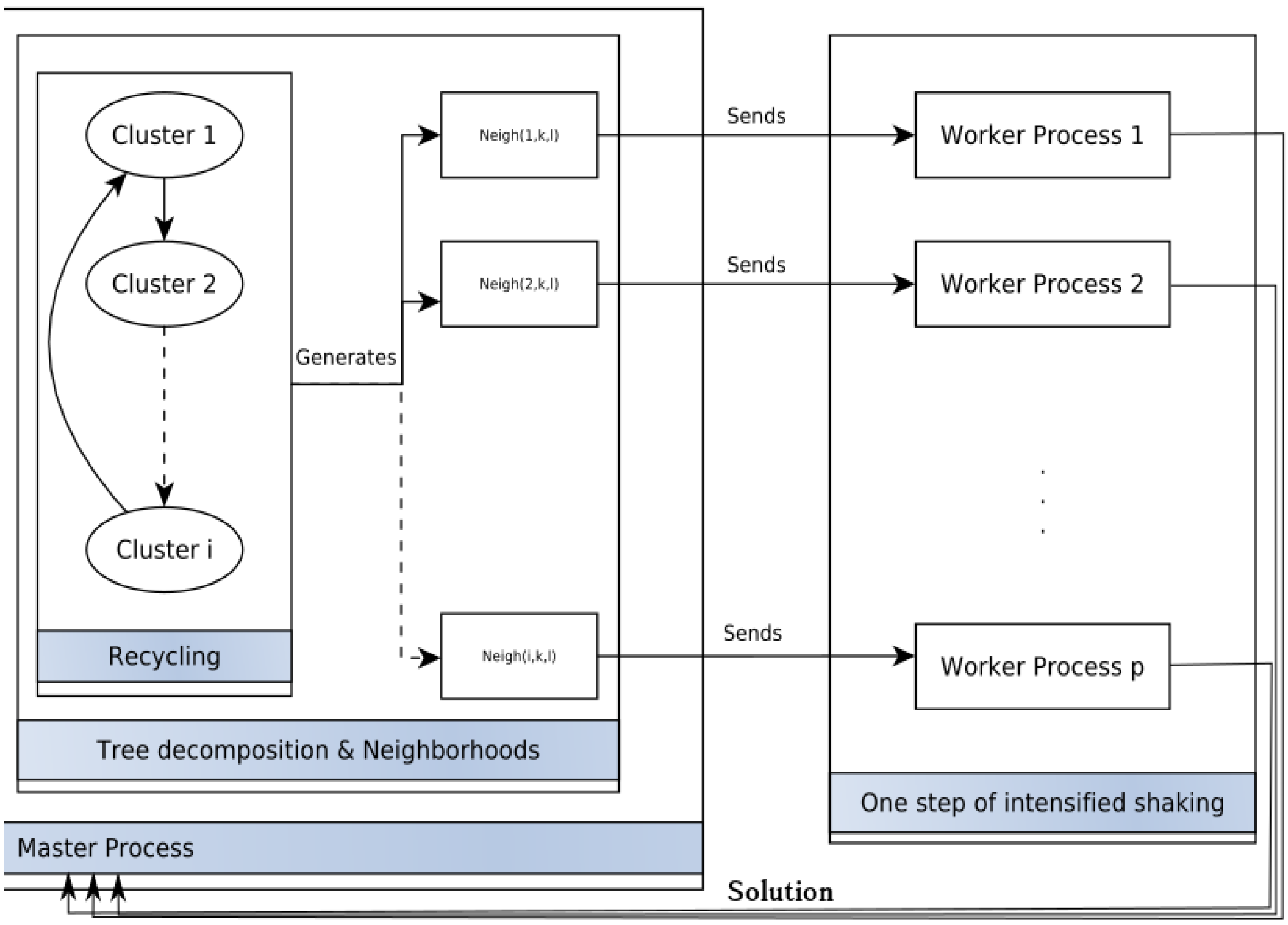


Results

(UAI17)



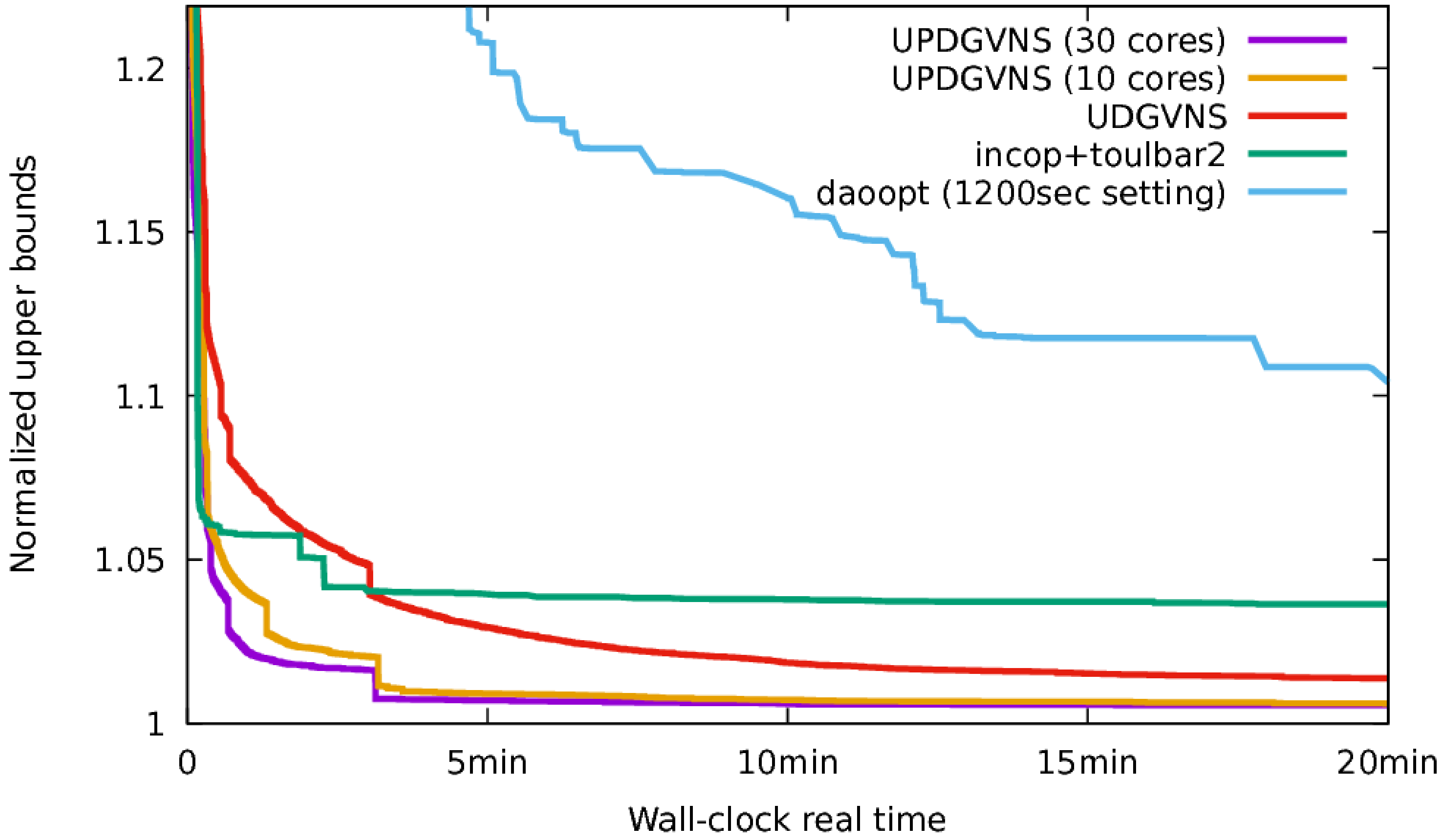
Parallel VNS



Unified Parallel Decomposition Guided VNS (UPDGVNS)

Results

(UAI17)



Bibliography

- For stochastic local search, see:
  ID Walk: a Candidate List Strategy with a Simple Diversification Device, B. Neveu, G. Trombettoni, F. Glover, CP 2004.

- For variable neighborhood search, see

Iterative Decomposition Guided Variable Neighborhood Search for Graphical Model Energy Minimization, Ouali et al., *UAI2017*.